FAULT IDENTIFICATION IN ROTATING MACHINERY USING NEURAL NETWORKS AND VIRTUAL INSTRUMENTATION

by

DHIROJ KUMAR PADHY

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

January, 2001

FAULT IDENTIFICATION IN ROTATING MACHINERY USING NEURAL NETWORKS AND VIRTUAL INSTRUMENTATION

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

January, 2001

by

DHIROJ KUMAR PADHY

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR – 208016 (INDIA)

CERTIFICATE

It is certified that the work contained in the thesis entitled "**Fault Identification in Rotating Machinery Using Neural Networks and Virtual Instrumentation**" by *Dhiroj Kumar Padhy* has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Nalinaksh Vyas.

Dr. Nalinaksh S Vyas
(Professor)
Department of Mechanical Engineering,
Indian Institute of Technology, Kanpur.

January 2001

ACKNOWLEDGEMENTS

I wish to express my profound gratitude and indebtedness towards my thesis supervisor Dr. N.S. Vyas for his inspiring guidance, invaluable suggestions and constant encouragement. His openness and grant of adequate freedom to me for thesis work kept my thought process unbridled. I am also thankful to Dr. V. Raghuram for his valuable suggestions, whenever needed.

I am extremely thankful to Animesh, Nivea, Sunil, Prashanth and Yogesh who devoted their valuable time and helped me in all possible ways towards successful completion of my thesis work. I appreciate and extend my thanks to Mr. J.P. Verma for his untiring help in conducting my experiments. I thank all those who have contributed directly or indirectly to my thesis.

I would like to thank all my friends for making my stay at IITK very enjoyable and memorable. I will cherish the moments forever.

I am also indebted to Indian Navy for having provided me the opportunity to take this course.

Indian Institute of Technology, Kanpur. Dhiroj Kumar Padhy

January, 2001.

ABSTRACT

Artificial Neural Networks offer an efficient platform for condition monitoring strategies in machinery and plants where the number of components and processes are too many and complex to be mathematically modeled appropriately. A machine diagnosis scheme, generally makes use of a knowledge based technique to identify the reasons for departure of the machine from its normal behavior. A condition monitoring activity is also intricately dependent on the quality of instrumentation and efficient data processing. The present study aims to integrate the fast emerging Virtual Instrumentation procedures with techniques of neural networks for fault identification in rotating machinery. The emphasis is on development of automated and user-interactive procedures for data acquisition as well as fault diagnosis. Experiments have been conducted on a Machinery Fault Simulator rig. Various types of faults were introduced and vibration data acquired from multiple stations on the rig were processed and employed to train the neural networks. The networks were based on back-propagation and probabilistic procedures. Neural Network training was carried out with the chosen architecture till desired degree of convergence is achieved. The network is finally tested and validated for test data.

CONTENTS

NOMENCLATURE

$\Delta_p w_{kg}^h$	Weight change at layer h
α	Momentum
β	Bearing load contact angle
δ_{pk}^h	Individual error for input vector p at node k in layer h
η	Learning rate
$\varphi(.)$	Activation function
θ_L^h	Threshold for node L of layer h
ρ	Constant
σ_i	Smoothing parameter
D_{ball}	Diameter of ball or roller
D_{pitch}	Pitch diameter of balls or rollers
E	Symptom
E_p	Mean square error
F_{bd}	Ball Spin defect frequency
F_{cd}	Cage defect frequency
F_{ird}	Inner race defect frequency
F_{ord}	Outer race defect frequency
H_i	i^{th} fault
K	Number of faults
M	Number of observations in training symptom pattern E
N	Number of contained balls or rollers
RPM	Rotational speed
f_j^h	Activation function at node j of hidden layer h

n_i	Number of training symptom patterns pertaining to i^{th} Fault
net^{h}_{pj}	Net input values to h^{th} hidden layer
o^{h}_{pk}	Output at output layer units
u_k	Output from summing function
v_k	Modified output from summing function
w_{kj}	Weight for synaptic connection from j^{th} node to k^{th} neuron
x_j	Input to network from j^{th} input node
x_{pN}	Non-weighted input to n^{th} neuron in p^{th} input vector
y_k	Output from neuron

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Rotordynamics has been an area of extensive research during the past two decades. Experimental and analytical techniques, both have been routinely used for diagnosing faults in operational rotating machinery. These techniques are also used during the design and development processes. Condition monitoring of rotating machinery is essential in order to ensure operation within design constraints and anticipate problems in time, so as to prevent complete failure. Advances in data acquisition techniques and the developments in the areas of knowledge acquisition, processing and programming offer potential for the application of automated strategies for machinery process supervision which would permit early detection of faults than is possible through conventional limits and trend checks. Meaningful and more efficient strategies for condition monitoring of rotating machinery are a pressing need, due to increasing demands on reliability and safety for power generating systems. Vibration analysis has proved to be a powerful tool for analysis of failure in rotating machinery. Vibration analysis with appropriate instrumentation can provide reliable knowledge of machine condition and allows relatively inexpensive repair on a pre-scheduled controlled basis in comparison to catastrophic failure whose repair may cost much more.

There are three phases in the evolution of on-line computer-based vibration monitoring of rotating machinery. The first phase includes collecting the vibration data and other operating parameters that are relevant. Phase two consists of identifying the complicated but normal behavior of the system, accounting for the variation of operating parameters. In phase three, use is made of knowledge based systems (expert systems) to identify the reasons for departures from normal behavior. Artificial Neural Networks (ANN) also find their role in the third phase of a monitoring strategy. Condition Monitoring implicitly involves data acquisition and analysis. Virtual Instrumentation is a recent technology that offers vast potential for easy and economic data acquisition and analysis.

The objective of the present study is to integrate techniques of Artificial Neural Networks with Virtual Instrumentation hardware and software to carry out experimental investigations on a Machinery Fault Simulator rig on fault identification. The Rig can be used to simulate a variety of machinery faults in a controlled manner. Vibration signals are picked up on the rig from multiple stations, using accelerometers or proximity pick-ups. The signals are conditioned and transferred to a computer. An attempt has been made to automate most of the signal processing activities by writing suitable codes in different software packages. Online data acquisition, time domain data display, Fast Fourier Transformation and frequency domain display was carried out using Virtual Instrumentation in LabVIEW. Feature extraction from the frequency domain data has been carried out in MATLAB. Data base creation, neural network training and testing has been carried out using Artificial Neural Network (ANN) toolbox available in MATLAB. Simulation and data acquisition has been illustrated for eighteen faults which include individual faults as well as in combination with one another. Out of these six cases were selected for development of the ANN scheme. Training is carried out, in each case with one hundred sets of vibration data from four channels. Back-Propagation and Probabilistic Neural Network (PNN) have been employed. Post training simulation for ten sets of vibration data for each fault showed a 100% correct diagnosis for both networks.

A brief review of commonly occurring faults in rotating machinery, their symptoms, and fault diagnosis procedures are discussed in Chapter 2. The basic principles and common algorithms of ANN are discussed in Chapter 3. Chapter 4 describes the experimental set-up and data acquisition. Feature extraction, network training and validation are covered in Chapter 5. Conclusions and scope for future work are listed in Chapter 6.

CHAPTER 2

FAULT DIAGNOSIS IN ROTATING MACHINERY

The commonly occurring faults in rotating machinery and currently available strategies for fault diagnosis and expert systems for fault diagnosis are discussed in this chapter.

2.1 Common Faults in Rotating Machinery

Rotordynamics has been an area of extensive research during the past two decades. Reference can be made to the studies by Sohre (1991), Rao (1992), Childs (1993), Dimentberg (1988) for a comprehensive discussion in the field, including fault classification and diagnosis. The most common problems are generally related to *ABC* category (Eisenmann, 1998) – Alignment, Balance and incorrect Clearances (typically on bearings). The most prominent types of rotating machinery problems are discussed below along with the causes and symptoms.

Unbalance

Rotor unbalance is the major problem in most rotating machinery. Unbalance forces arise due to eccentricity between the mass and geometric centers caused by material, manufacturing defects or by damage during operation. Vibration due to unbalance prominently occurs at a frequency equal to the rotor speed. However, specific unbalance forces caused as in the case of ball bearing wear give rise to additional characteristics like high frequency signals corresponding to bearing vibration frequency.

Misalignment

Misalignment is present in almost every machine to some extent. During the course of operation a small degree of misalignment may attain higher proportions. Misalignment can be classified as parallel or angular. It is a major source of vibration and leads to other problems like bearing overheat/damage, loss of power etc. Frequencies are

characteristically twice the running speed, but first and sometimes third or fourth multiples of running speed are also observed.

Rolling Element Bearing Defects

Rolling element bearings contain elements that roll between an inner race and an outer race with very close clearances. They provide high stiffness to the rotor but are low in damping. Application of high loads, shock loads, or bearing attrition due to extended run times increase the internal clearances resulting in bearing failure. Typically four types of defects are observed in rolling element bearings, each giving rise to vibrations at a specific frequency. Reference can be made to Eisenmann (1998) for a detailed discussion on bearing problems. Briefly the faults and the predominant vibration frequencies can be summarized as

Type of Defect	Frequency	
Outer Race Defect	$F_{ord} = \left\{\frac{N}{2}\right\} \times \left\{1 - \left(\frac{D_{ball}}{D_{pitch}}\right) \times Cos\beta\right\} \times RPM$	(2.1)
Inner Race Defect	$F_{ird} = \left\{\frac{N}{2}\right\} \times \left\{1 + \left(\frac{D_{ball}}{D_{pitch}}\right) \times Cos\beta\right\} \times RPM$	(2.2)
Ball Spin Defect	$F_{bd} = \left(\frac{1}{2}\right) \times \left\{\left(\frac{D_{pitch}}{D_{ball}}\right) - \left(\frac{D_{ball}}{D_{pitch}}\right) \times (Cos\beta)^2\right\} \times RPM$	(2.3)
Train/Cage Defect	$F_{cd} = \left(\frac{1}{2}\right) \times \left\{1 - \left(\frac{D_{ball}}{D_{pitch}}\right) \times Cos\beta\right\} \times RPM$	(2.4)

where,

N = Number of contained balls or rollers

RPM = Rotational speed of bearing inner race (Hz)

D_{ball} = Diameter of ball or roller

D_{pitch} = Pitch diameter of balls or rollers

β = Bearing load contact angle (degrees)

Cocked Rotor

A rotor with its plane not perpendicular to axis of rotation is termed cocked. Also, unevenness in mass distribution on either side of the plane perpendicular to rotational axis and passing through geometric center of the rotor produces cockiness. This fault generates excessive vibration in axial direction prominently at the first multiple of rotational frequency. Sometimes, second or third multiples of rotational frequency are also observed.

Rotor Rub

Contact of any mechanical element of the system with the rotor causes rubbing. It can be radial or axial. Mechanical seal rub is a typical example of radial rub and casing rub at shaft end is a case of axial rub. Mostly both the types of rubs generate vibration at frequency equal to twice the rotational frequency but lower and higher frequencies are also common.

Distortion

Casing and foundation distortion cause vibration in an indirect way, either by generating misalignment between driver and driven machines or by causing internal rub or uneven bearing contact. This in turn transmits forces to the rotor, inducing it to generate forces of its own, such as unbalance and a wide variety of oil film and friction induced forces. Another possibility is that the loads on casing supports shift and can set off a series of resonance problems. Piping forces and foundation distortion often cause this type of difficulty.

Asymmetric Shaft

The response of the asymmetric shaft has several harmonics and the frequencies observed are first, second or third multiples of the rotational frequency.

Mechanical Looseness

Loose rotor components such as disks, sleeves, thrust collars etc. cause internal friction problems. The frequency of vibration is always the rotor critical speed. Mechanical loose

components like bolts give rise to $1 \times rev$ and harmonic frequency signals due to secondary phenomena. The amplitude and phase continually change. Loose assembly of bearings give rise to sub-harmonic response and the typical frequency response is $0.5 \times rev$ and $(1/3) \times rev$. This could be mistaken for oil film whirl, particularly in the region of twice the rotor critical speed, around the threshold of instability.

Oil Film Whirl

Oil whirl in an oil lubricated journal bearing can occur due to preload forces, shaft/bearing conditions, shaft eccentricity/concentricity or initial rotor deflection. In such an eventuality a flowing wedge of oil forms in the bearing and drives the shaft ahead of its forward circular motion within the bearing clearance. Theoretically, the frequency of response of oil whirl should be at $0.5 \times rev$. However, it occurs around $0.45 - 0.48 \times rev$.

Bending Resonance

Bending critical of the rotor occurs when the rotational speed is equal to its lateral resonance frequency. Resonance of the structure, support and auxiliaries cause fairly large amplitudes of vibration at the rotor speed and this occurs over a narrow range of speed of operation. Such resonance can be cured by the addition of damping.

Gearbox Defects

Gearboxes generate additional frequency components equal to running speeds of various shafts associated with them. Also, gear teeth meshing frequencies are prominent. Defects in any of the shafts show frequency components as multiples of the rotational frequency of that shaft. Gear teeth damage/break off show higher amplitudes at gear teeth meshing frequency.

A summary of the rotating machinery defects, characteristics and common remedial measures are given in Table 2.1.

Table 2.1 Identification and Correction of Malfunctions in Rotating Machinery

Sl. No	Fault	Frequency	Spectrum; Time-domain, Orbit shape	Correction
1	Mass unbalance	n	Distinct n with much lower values of 2n, 3n etc. Elliptical and circular orbits	Field or shop balancing
2	Misalignment	n, $2n$ etc.	Distinct n with equal or higher values of $2n$, $3n$ etc.	Hot and/or cold alignment
3	Shaft bow	n	Drop out of vibration around critical speed in Bode plot	Heating or peening to straighten rotor
4	Steam loading	n	Load Sensitive n	Modify admission sequence, repair diaphragms, install nozzle blocks properly
5	Bearing wear	n, subharmonics, orders	High n, high $0.5n$, sometimes $2n$, very high frequencies, can not be balanced	Replace bearing
6	Cracked rotor	n, $2n$	High n, 0.5 x critical speeds may show on coast down	Remove rotor
7	Gravity excitation	$2n$	0.5 x critical speed appears on Bode plot	Reduce eccentricity by balancing
8	Looseness	N plus large number of orders, $0.5n$ may also show up	High n with lower-level orders, large 0.5 orders	Tighten shim and bolts
9	Coupling lockup	n, $2n$, $3n$ etc.	n with high $2n$, similar to misalignment: Starts and stops may yield different vibration patterns	Replace coupling or remove sludge
10	Thermal instability	n	n has varying phase angle and amplitude	Compromise balance or remove problem
11	Oil-whirl	$0.4n$ to $0.48n$	Subsynchronous component less than 0.5 order. Informal loop in orbit	Temporary: load bearing heavier-correct misalignment; long term: change bearing
12	Oil-whip	p_1	Subsynchronous component does not change with speed	Change bearing type
13	Subharmonic response	$0.5n$, $1/3\ n$, $0.25n$ and higher	Subsynchronous vibration depending on natural frequencies	Remove looseness, excessive flexibility. Alter natural frequency
14	Rubs	$0.25n$, $1/3n$, $0.5n$ or orders	External loops in orbit	Eliminate conditions such as thermal bow and mass unbalance that cause rub
15	Hysteresis	$0.65n$ to $0.85n$	High magnitude fractional frequency (greater than $0.5n$)	Eliminate or secure built-up parts

Note: n – Running speed

p_1 – First natural frequency

2.2 Diagnostic Techniques

Experimental and analytical techniques, both, have been routinely used for diagnosing faults in operational rotating machinery. These techniques are also used during the design and development processes. Parameter identification techniques also find wide use to quantify natural frequencies, mode shapes, damping stability and frequency-response characteristics. Some of these diagnostic techniques are discussed below.

2.2.1 Time History or Waveform Analysis

Data is sampled at specific time intervals and plotted as a function of time. This gives a realistic view of the actual dynamic behavior of a machine. However, the data is normally too difficult to analyze in such form as signals are not always periodic and are mostly polluted by noise. Noise can be reduced by signal conditioners but cannot be eliminated altogether. An important feature of waveform analysis is its ability to interpret non-steady conditions as observed in compressors, pumps, hydraulic motors, machine tools etc. It is possible to extend the use of waveform analysis to study of phase changes and short transient impulses. Valuable information is provided by signal shape i.e. truncation, pulses, modulation, glitch or shaft-induced signals obtained from a proximity probe that are caused by scratches on the shaft. Time between events represents the frequency component specific to the machine. The phase between two signals provides information about vibratory behavior that can be used to diagnose a fault such as a misalignment and mass unbalance.

2.2.2 Orbital Analysis

Signals obtained from two probes mounted perpendicular to each other in a single vertical plane perpendicular to the shaft, when plotted on two mutually perpendicular axes, generate an actual trace of a shaft's rotation, called the *orbit*. Orbital analysis is employed to diagnose problems like shaft pre-loading. For example, a horizontal pre-load (e.g. misalignment) restricts horizontal motion and will produce an elliptical orbit with the vertical axis longer than the horizontal. Other defects like torque reaction, pressure angle forces associated with gearboxes, internal seal rub due to casing distortion, shaft rubs etc. can also be studied through orbital analysis.

2.2.3 Spectrum Analysis

Spectrum analysis is conducted with an FFT algorithm or filters. The processed signal gives information about the amplitude and phase content at various frequencies. Frequencies of vibration response can be related to direct excitation frequencies or their orders, natural frequencies, sidebands, subharmonics and sum-difference frequencies. Frequencies of the measured vibration response are equal to those of the forces causing the vibration. The operating speed of the machine is usually the fundamental frequency used in the analysis. The peaks at multiple orders of the fundamental are due to a variety of faults that are present in the rotating machinery. The exact fault can be predicted by comparing with good test data that is directly proportional to the information available on the design of a machine and its working mechanisms.

2.2.4 Cepstrum Analysis

Cepstrum is defined as the inverse Fourier transform of the logarithmic power spectrum commonly used in vibration analysis. The complex cepstrum is reversible to a time signal. A cepstrum is effective for accurately measuring frequency spacing, harmonic and sideband patterns in the power spectrum. One component of the cepstrum represents the global power content of an entire family of harmonics or sidebands. The severity of a defect such as that in a rolling element bearing is provided by one component of the cepstrum display.

2.2.5 Expert Systems

Expert systems are programs that use inference techniques involving formal reasoning normally performed by human experts. These systems can advice, diagnose, analyze and categorize based on previously defined knowledgebase. The system usually consists of two parts, the data or knowledge base and the interference engine, which is the diagnosis and advisory part. Usually the knowledge base consists of two parts, viz., the working memory and the rule-base. The rule-base does not change during a given consultation; thus it is a static database. The working memory consists of facts related to a specific consultation and it is thus a dynamic database. The dynamic memory is empty at the beginning of a diagnosis. During the diagnosis process, the interference engine uses the

facts and rules of the rule-base in conjunction with the data from the observed signals, to add facts in the working memory. Fig. 2.1 shows the principles of such an expert system program (Rao,1996).

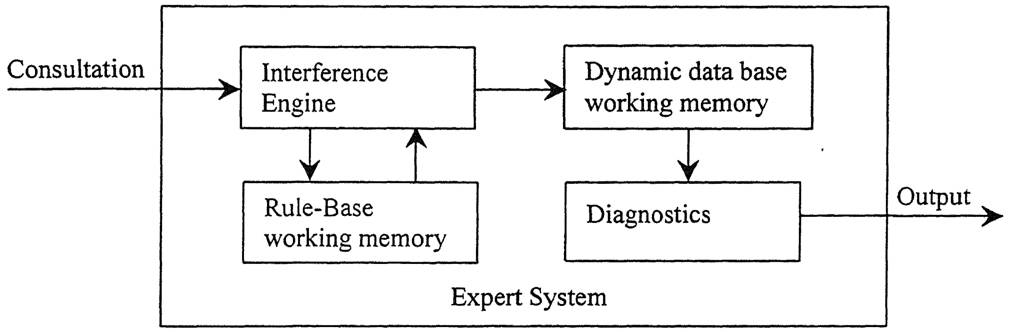


Figure 2.1 Structure of an Expert System

The work carried out in this thesis is based on Artificial Neural Network techniques, which are described in the next chapter.

CHAPTER 3

ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks are inspired by the functioning of the human brain. A human brain consists of millions of computing elements called *neurons*. They communicate through a connection network of axons and synapses. Similarly, computation is carried out in an ANN through several nodes called neurons each interconnected with the others. Similar to the human brain ANNs learn from experience, generalize from previous examples to new ones and abstract essential characteristics from the inputs containing randomness and irrelevant data.

ANN has been attracting the attention of scientists and technologists for long. McCulloh and Pitts formulated the first formal definition of a synthetic neuron model based on biological model in 1943. However, neural networks being massively parallel distributed processor, require tremendous amount of computation. Non-availability of fast and affordable computing power had limited its growth till the last decade of twentieth century. Application of ANNs to vibration and fault diagnosis are relatively few. Mayes (1994) applied ANN for on-line vibration monitoring of large turbo-generators. The investigations focussed on data processing and the use of neural networks. Elkordy, Chang and Lee (1994) investigated the applicability of ANNs for vibration signature analysis of a five-storey structure. McCormick and Nandi (1997) contemplated the application of ANN for real-time fault classification of rotating shafts. Vyas et. al. (2000) used Sohre's knowledge base to train one each of Back-propagation and Probabilistic Neural Networks to identify rotor faults.

3.1 The Artificial Neuron

The artificial neuron was developed to mimic the first order characteristics of the biological neuron. An artificial neuron is an information processing unit that is

fundamental to the operation of a neural network. A set of inputs is applied, each representing the output of another neuron. Each input is multiplied by a corresponding weight, analogous to a synaptic strength, and all of the weighted inputs are then summed to determine the activation level of the neuron. Fig. 3.1 shows a typical model of a neuron. The three basic elements of the neuron model are described below (Zurada, 1999).

Synapses: Synapses are connecting links, each of which is characterized by a weight or strength of its own. It specifies the connection between a signal x_j at the input of the sample *j*, and a neuron *k*. The weighting factor is w_{kj}.

Adder: Adder sums up the input signals weighted by the respective synapses of the neuron. The operation is similar to that of a linear adder.

Activation Function: An activation function defines the output of a neuron in terms of the activity level at its input. It limits the amplitude of the output of the neuron and introduces non-linearity into the network. The activation function is also referred to as *squashing function* since it is used to limit the output values of a neuron in a definite small range irrespective of the inputs to it.

The neuron also includes an externally applied threshold function, θ_K, which has the effect of lowering the net input of the activation function. On the other hand, employing a bias term rather than a threshold may increase the net input of the activation function. The bias is the negative of the threshold. In mathematical terms a neuron *k* is described by

$$u_k = \sum_{j=1}^{p} w_{kj} x_j \,; \qquad y_k = \varphi(u_k - \theta_k) = \varphi(v_k) \tag{3.1}$$

where, $x_1, x_2 \cdots x_p$ are the input signals, $w_{k1}, w_{k2} \ldots w_{kp}$ are the synaptic weights of the neurons *k*, u_k is the linear combiner output, θ_K is the threshold, $\varphi(.)$ is the activation

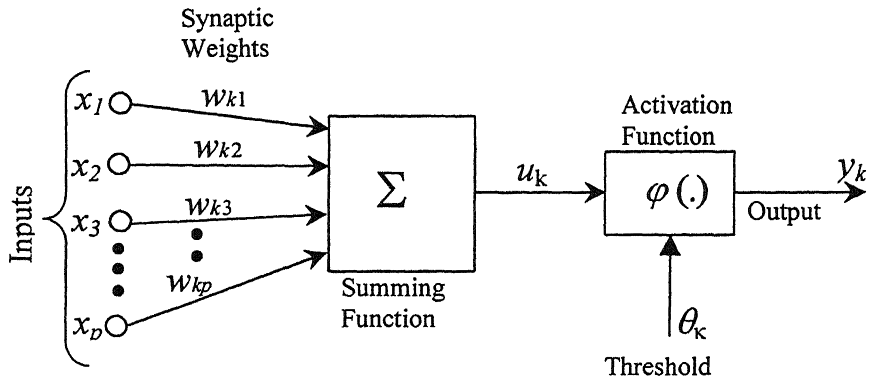


Figure 3.1 Model of a Nonlinear Neuron

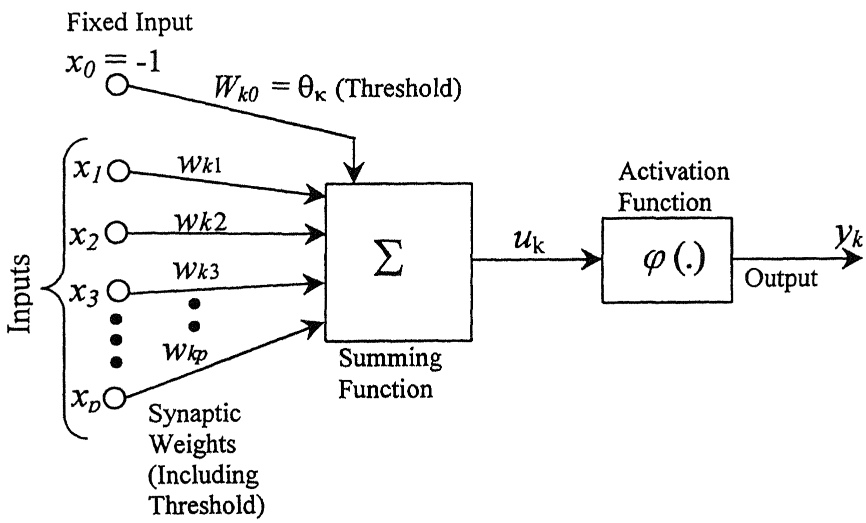


Figure 3.2 Modified Model of a Nonlinear Neuron

function and y_k is the output signal of the neuron. v_k is the output from the neuron after summation. Equation (3.1) can be equivalently written as

$$v_k = \sum_{j=0}^{p} w_{kj} \tag{3.2}$$

$$y_k = \varphi(v_k) \tag{3.3}$$

In equation (3.2) a new synapse has been added, whose input and weight, respectively, are

$$x_0 = -1 \ ; \qquad w_{k0} = \theta_k \tag{3.4}$$

This modified model of a neuron is shown in Fig. 3.2.

3.2 Network Architectures

The manner in which the neurons in a neural network are structured is intimately linked with the learning algorithm used to train the network. Four general classes of network architectures are:

(a) Single Layer Feed Forward Network: This is a network of neurons organized in the form of layers. The simplest form of a network consists of an input layer of source nodes that project on to an output layer of neurons, but not vice versa. This is of feed forward type (Fig. 3.3).

(b) Multi-layer Feed Forward Network: It is similar to the one described in (a) above except that it has one or more hidden layers of neurons. Hidden layers of neurons are interface between the input and output layers. It is observed that some problems converge better when these hidden layers are used (Fig. 3.4).

(c) Recurrent Networks: A recurrent network distinguishes itself from the feed-forward network in that it has at least one feedback loop. For example, a recurrent network may consist of single layer of neurons, with each neuron feeding its output signal back to the input of all other neurons (Fig. 3.5). A time delay may be introduced in

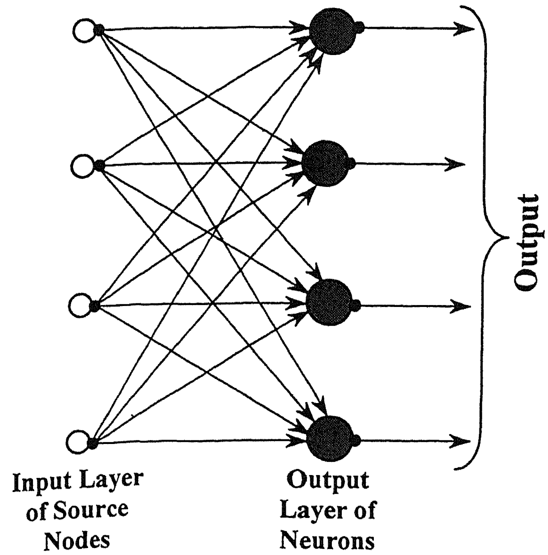


Figure 3.3 Single Layer Feed Forward Network

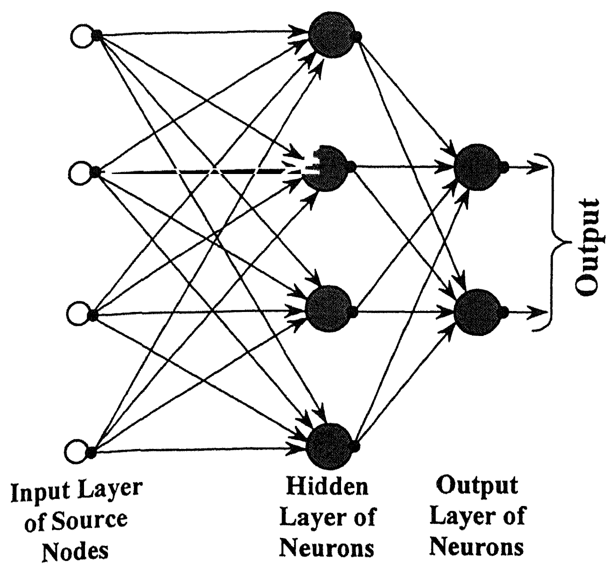


Figure 3.4 Feed Forward Network with One Hidden Layer

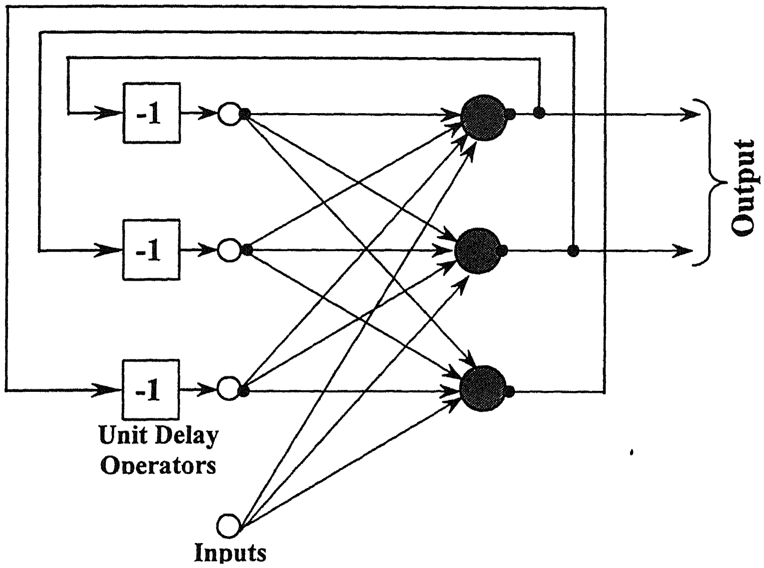


Figure 3.5 Recurrent Network

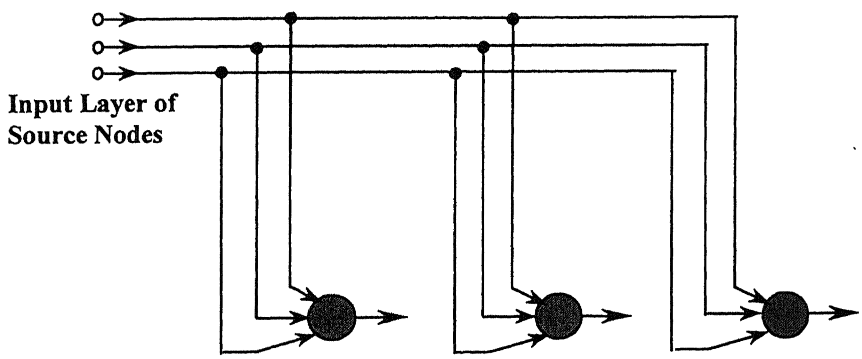


Figure 3.6 One Dimensional Lattice of 3 Neurons

the feedback path. Recurrent networks typically operate with a discrete representation of data and employ neurons with a hard-limiting activation function.

(d) Lattice Structure: A lattice consists of a one or more multi-dimensional array of neurons with a corresponding set of source nodes that supply the input signals to the arrays (Fig. 3.6). The dimension of the lattice refers to the number of dimensions in which the graph lies.

3.3 The Back-propagation Training Algorithm

The Back-propagation Training Algorithm (BPA) is a popular training algorithm. Its major attraction is its suitability to a large number of applications and high rates of convergence. It is designed to solve the problems of choosing weight values for layered artificial neural networks with feed forward connections from input layer to hidden layer and then to the output layer. The algorithm performs the input to output mapping by minimizing a cost function using a gradient search technique. The cost function, which is equal to the mean squared difference between the desired and the actual net output, is minimized by making wide connection adjustments according to the error between the computed and target output processing element values. There are two stages in the development of a back-propagation algorithm, namely forward pass and backward pass. During the forward pass all the weights of the network are initialized randomly and the network outputs and the difference between the actual and target output (i.e. the error) is calculated for the initialized weights. During the backward step, the initialized weights are adjusted to minimize the error by propagating the error backwards. The network outputs and error are calculated again with the updated weights and the process repeats till the error is acceptably small. These two steps are described below and the whole algorithm can be represented schematically as shown in Fig. 3.7.

Forward Pass

Referring to Fig. 3.7 the pth input vector to the network can be described as

$$x_p = \left(x_{pl} \; \ldots . x_{pN}\right)' \qquad (3.5)$$

where x_{pi} represents the input attribute i for the vector p.

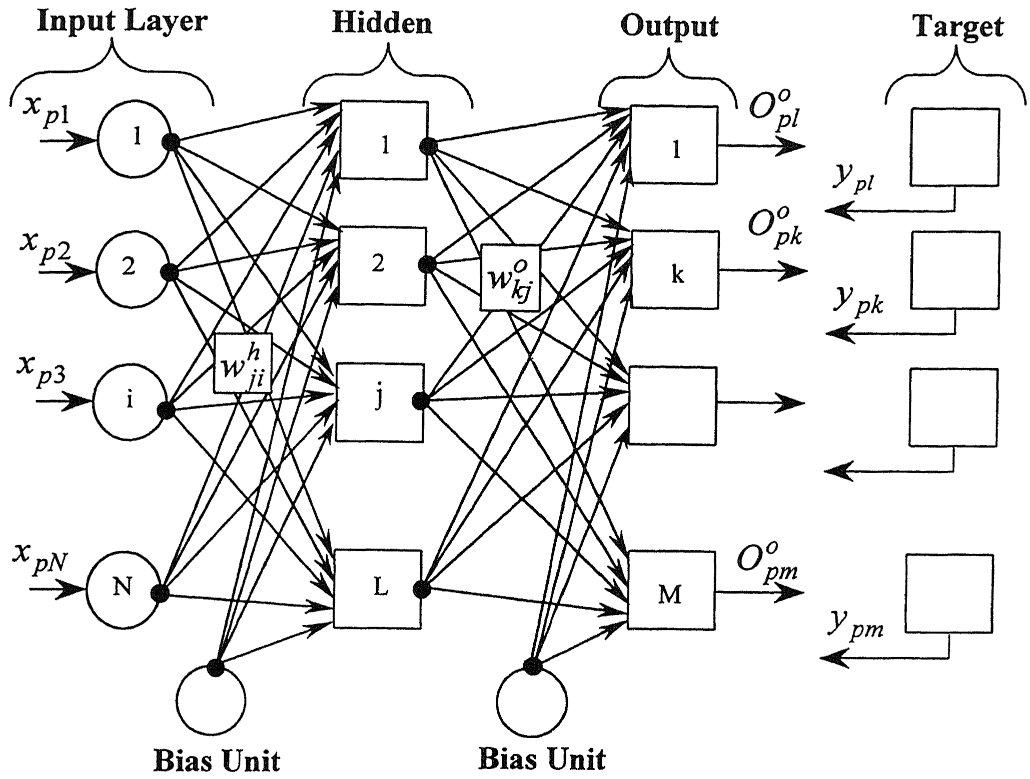


Figure 3.7 Three Layer Back-propagation Network Architecture

The net input to the hidden layer then becomes

$$net^h_{pj} = \sum_{i=1}^{N} w^h_{ji} x_{pi} + \theta^h_j \tag{3.6}$$

where, w^h_{ji} represents the weight of the layer h from node i to node j and θ^h_L represents the threshold for the node L of the layer h.

The outputs from the hidden layer (which is the input to the output layer) are

$$o^h_{pj} = i_{pj} = f^h_j\left(net^h_{pj}\right) \tag{3.7}$$

o^h_{pj} is the output from the node j of the hidden layer h, f^h_j is the activation function at node j of the hidden layer h. The equations (3.6) and (3.7) are represented graphically in Fig. 3.8.

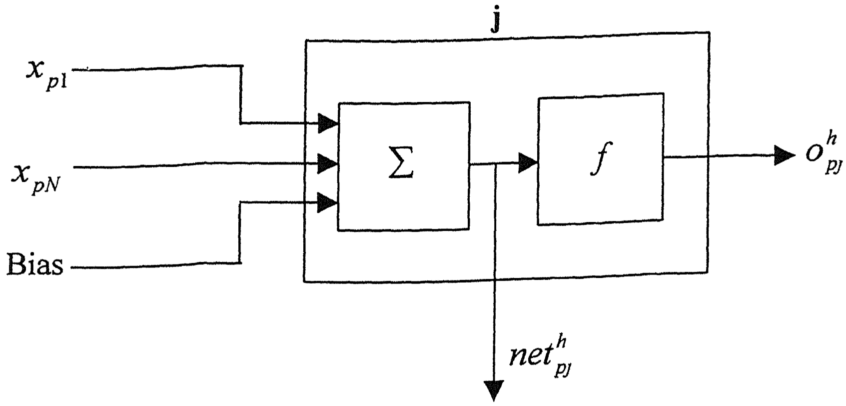


Figure 3.8 Output Calculation at Each Node in the Forward Pass

In the above the net-input values at output layer unit are

$$net^o_{pk} = \sum_{j=1}^{L} w^o_{kj} i_{pj} + \theta^o_k \tag{3.8}$$

while the outputs at output units are

$$o^o_{pk} = f^o_k\left(net^o_{pk}\right) \tag{3.9}$$

Individual error at each output unit is

$$\delta_{pk} = y_{pk} - o^o_{pk} \tag{3.10}$$

from which the overall mean square error can be computed as

$$E_p = \tfrac{1}{2} \sum_{k=1}^{m} \delta_{pk}{}^2 \tag{3.11}$$

Backward Pass

Weight adjustment is carried out at the output layer through the following procedure. Using equations (3.10) and (3.11), the mean square error, E_p, can be expressed as

$$E_p = \tfrac{1}{2}\Sigma\left(y_{pk} - o^o_{pk}\right)^2 = \tfrac{1}{2}\Sigma\left\{y^o_{pk} - f^o_k\left(net^o_{pk}\right)\right\}^2 \tag{3.12}$$

The weight change of an output layer weight is the negative gradient of E_p with respect to output layer weights w^o_{kj} and can be written as

$$\frac{\partial E_p}{\partial w^o_{kj}} = -\left(y_{pk} - o^o_{pk}\right)\frac{\partial f^o_k}{\partial\left(net^o_{pk}\right)}\frac{\partial\left(net^o_{pk}\right)}{\partial w^o_{kj}} \tag{3.13}$$

However, from equation (3.8)

$$\frac{\partial\left(net^o_{pk}\right)}{\partial w^o_{kj}} = \frac{\partial}{\partial w^o_{kj}}\left(\Sigma\, w^o_{kj} i_{pj} + \theta^o_k\right) = i_{pj} \tag{3.14}$$

and from equation (3.9)

$$\frac{\partial f^o_k}{\partial\left(net^o_{pk}\right)} = f^{o\prime}_k\left(net^o_{pk}\right) = o^o_{pk}\left(1 - o^o_{pk}\right) \tag{3.15}$$

Therefore, the weight change at the output layer weight is

$$\Delta^o_p w = \frac{-\partial E_p}{\partial w^0_{kj}} = \left(y_{pk} - o^o_{pk}\right) f^{o\prime}_k\left(net^o_{pk}\right) \times i_{pj} \tag{3.16}$$

Now denoting

$$\delta^o_{pk} = \left(y_{pk} - o^o_{pk}\right) f^{o\prime}_k\left(net^o_{pk}\right) \tag{3.17}$$

the weight change at the output layer weights can be written as

$$\Delta^o_p w = \delta^o_{pk} i_{pj} \tag{3.18}$$

To make the learning process smooth and to ensure that the weight changes take place in the same direction, two network parameters – learning rate coefficient η and momentum α. – are introduced in lieu of direct application of the above mentioned weights, so that

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \eta\left(y_{pk} - o_{pk}^{o}\right) f_{k}^{o'}\left(net_{pk}^{o}\right) \times i_{pj} + \alpha w_{kj}^{o}(t-1) \qquad (3.19)$$

A small value of η implies that the network will have to make a large number of iterations. Its value is normally kept between 0.05 and 0.9. It is often possible to increase its value as the network error decreases, thereby increasing the speed of convergence. Another way to increase convergence speed is by adopting an extra momentum term while updating the weights. This additional term tends to keep the weight changes in the same direction. The entire weight updating process at the output layer can be represented as shown in Fig. 3.9.

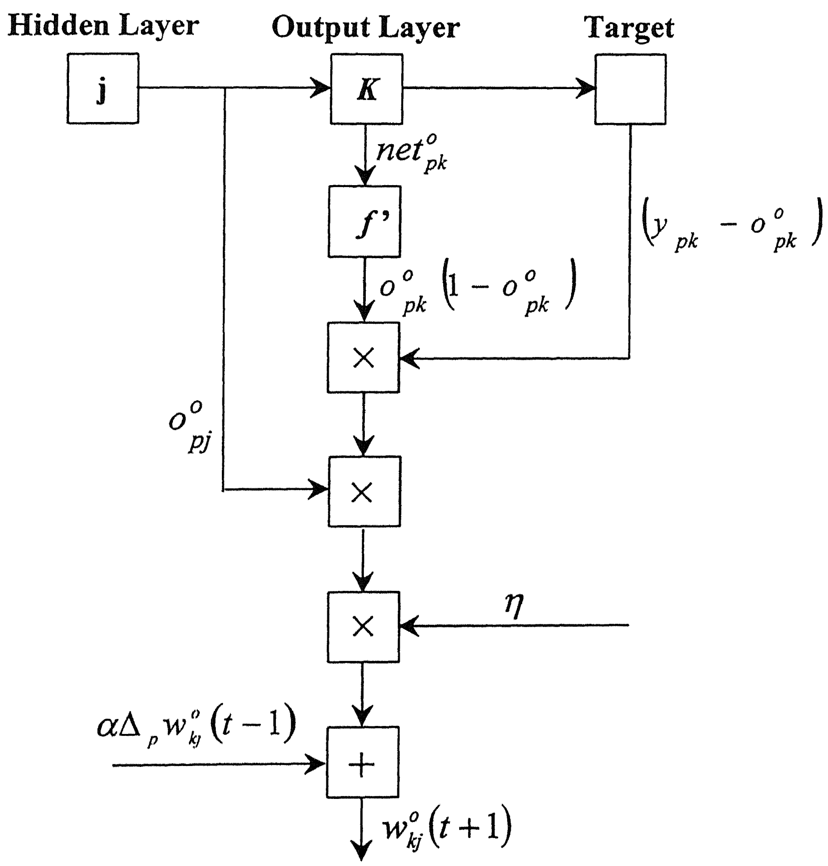


Figure 3.9 Output Layer Weight Updating

While updating the weights for the hidden layers it should be noted that there is no target output and therefore the adjustment of weights is proportional to their initial contribution. From equations (3.8) and (3.9) one gets

$$E_p = \tfrac{1}{2}\sum_p \left\{ y_{pk} - f_k^o\left(\sum w_{kj}^o i_{pj} + \theta_k^o\right)\right\}^2 \tag{3.20}$$

The weight change of hidden layer weights is the negative gradient of E_p with respect to hidden layer weights w_{ji}^h and is given by

$$\frac{\partial E_p}{\partial w_{ji}^h} = -\sum_k \left(y_{pk} - o_{pk}^o\right)\frac{\partial o_{pk}^o}{\partial(net_{pk}^o)}\frac{\partial\left(net_{pk}^o\right)}{\partial i_{pj}}\frac{\partial i_{pj}}{\partial(net_{pj}^h)}\frac{\partial\left(net_{pj}^h\right)}{\partial w_{ji}^h} \tag{3.21}$$

and the individual terms on the right hand side of the above equation can be expanded as

$$\frac{\partial o_{pk}^o}{\partial(net_{pk}^o)} = \frac{\partial(f_k^o(net_{pk}^o))}{\partial(net_{pk}^o)} = f_k^{o'}\left(net_{pk}^o\right) = o_{pk}^o\left(1-o_{pk}^o\right) \tag{3.22}$$

$$\frac{\partial\left(net_{pk}^o\right)}{\partial i_{pj}} = \frac{\partial\left(\sum_{j=1}^{L} w_{kj}^o i_{pj} + \theta_k^o\right)}{\partial i_{pj}} = w_{kj}^o \tag{3.23}$$

$$\frac{\partial i_{pj}}{\partial(net_{pj}^h)} = \frac{\partial\left(f_j^h net_{pj}^h\right)}{\partial(net_{pj}^h)} = f_j^{h'}(net_{pj}^h) = o_{pj}^h\left(1-o_{pj}^h\right) \tag{3.24}$$

$$\frac{\partial\left(net_{pj}^h\right)}{\partial w_{ji}^h} = \frac{\partial\left(\sum_{i=1}^{N} w_{ji}^h x_{pi} + \theta_j^h\right)}{\partial w_{ji}^h} = x_{pi} \tag{3.25}$$

Substituting equations (3.22)-(3.23) in equation (3.21) one gets

$$\Delta_p w_{ji}^h = -\frac{\partial E_p}{\partial W_{ji}^h} = x_{pi} o_{pj}^h \left(1 - o_{pj}^h\right) \sum_k \left(y_{pk} - o_{pk}^o\right) o_{pk}^o \left(1 - o_{pk}^o\right) w_{kj}^o \qquad (3.26)$$

Network parameters η and α can be introduced in a manner similar to that in the case of the output layer, to express the final weight change at the hidden layer as

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \eta x_{pi} o_{pj}^h \left(1 - o_{pj}^h\right) \sum \left(y_{pk} - o_{pk}^o\right) o_{pk}^o \left(1 - o_{pk}^o\right) w_{kj}^o + \alpha w_{ji}^h(t-1) \qquad (3.27)$$

The entire weight updating process at hidden layer can be schematically represented as shown in Fig. 3.10.

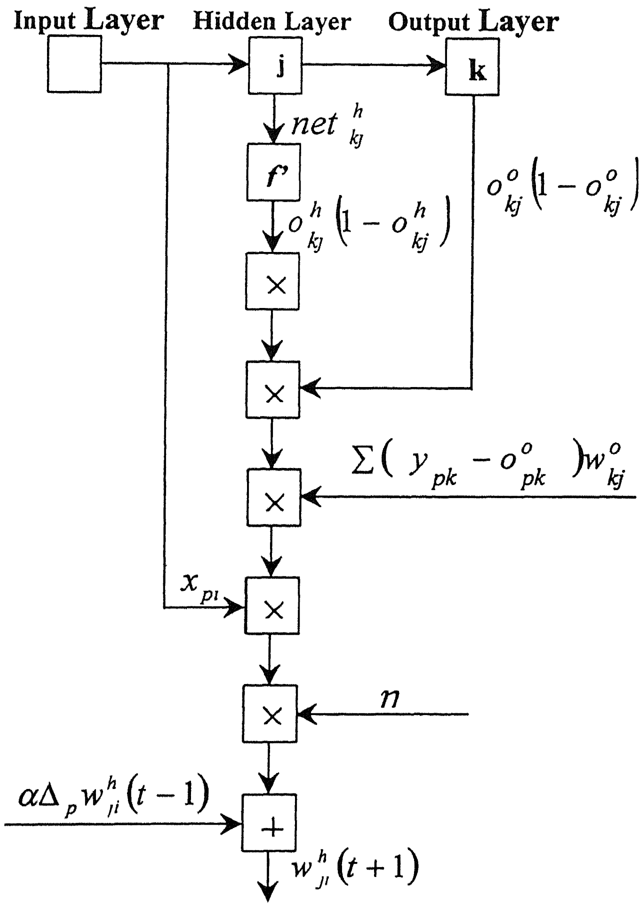


Figure 3.10 Hidden Layer Weight Updating

3.4 Types of Activation Functions

Various types of transfer functions, $\varphi(.)$, are employed in developing a neuron. Some of the common transfer functions are described below.

Name	Mathematical Representation
• Linear Activation Function	$\varphi(x) = x$
• Positive Linear Activation Function	$\varphi(x) = x \quad \text{for } x \geq 0,$ $= 0 \quad \text{for } x \leq 0.$
• Hyperbolic Linear Tangent Sigmoid Activation Function	$\varphi(x) = \dfrac{2}{1+\exp(-\rho x)} - 1$
• Logistic Sigmoid Activation Function	$\varphi(x) = \dfrac{1}{1+\exp(-\rho x)}$

where ρ is a constant, and $\varphi(.)$ is the activation function. The above four activation functions are represented graphically in Fig. 3.11.

3.5 Training Algorithms

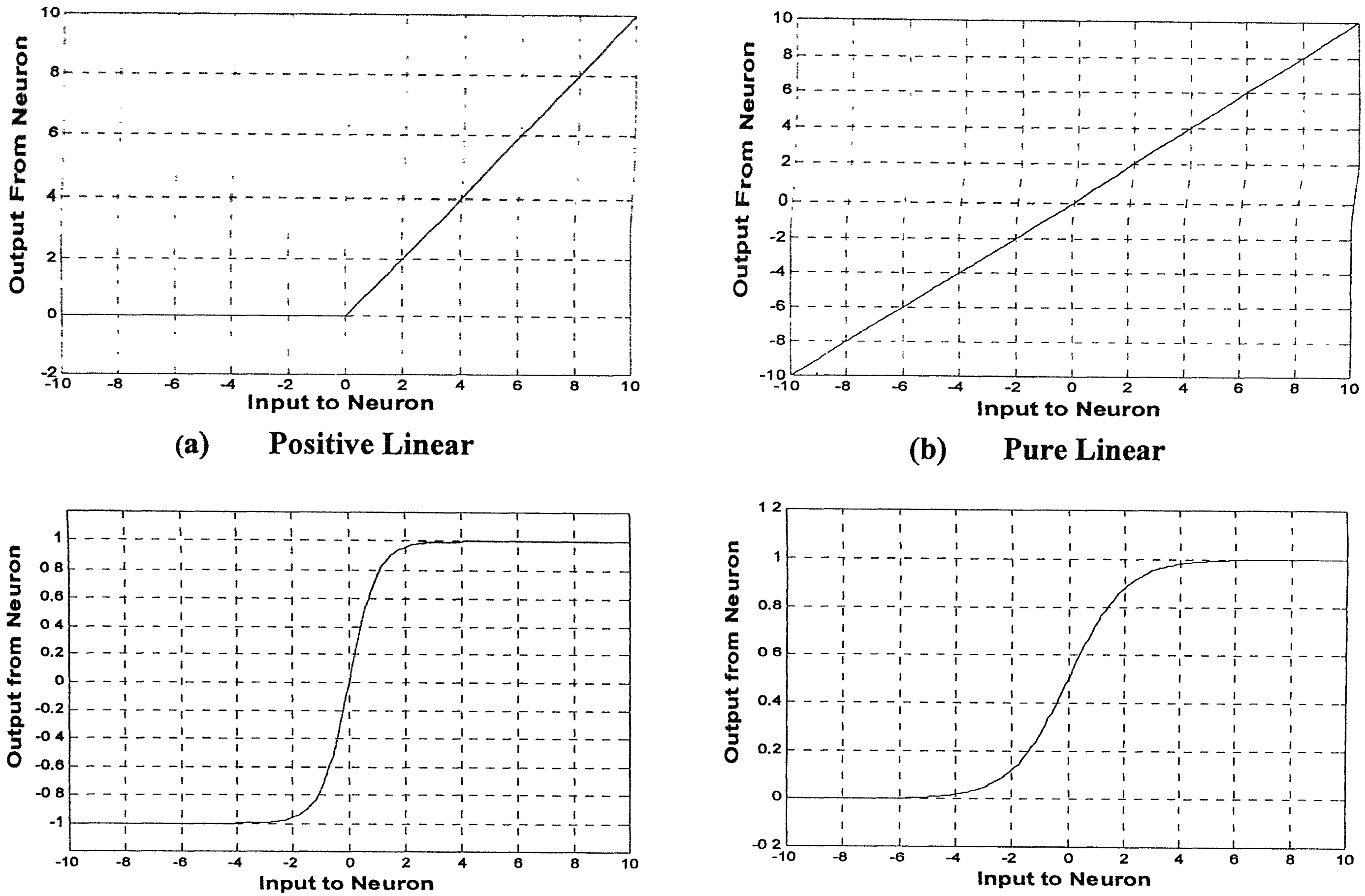
Training is accomplished in a network by sequentially applying input vector, while adjusting network weights according to a predetermined procedure. During training, the network weights gradually converge to values such that each input vector produces the desired output vector. Neural network toolbox in MATLAB provides fifteen different learning functions. Out of these two were used in the current study and are briefly described below.

Gradient Descent Back-propagation ('TRAINGD')

TRAINGD updates weight and bias values according to gradient descent as follows

$$\mathrm{d}w = \eta \times \mathrm{d}E_p / \mathrm{d}w \tag{3.28}$$

where, w is the weight/bias variable, η is the learning rate, and E_p is the performance.



(a) Positive Linear

(b) Pure Linear

(c) Hyperbolic Tangent Sigmoid

(d) Logistic Sigmoid

Figure 3.11 Activation Functions

Resilient Back-propagation ('TRAINRP')

This training algorithm updates weight and bias values according to the resilient back-propagation algorithm as follows

$$dw = \Delta w \times sign(gw) \tag{3.29}$$

where, the elements of Δw are all initialized to $\Delta 0$ and gw is the gradient. At each iteration the elements of Δw are modified. If an element of gw changes its sign between one iteration and the next, then the corresponding element of Δw is decreased by certain value. If it maintains the same sign from one iteration to the next, then the corresponding element of Δw is increased by certain value. This continuos change in value of Δw ensures a gradual but faster convergence.

3.6 Probabilistic Neural Network (PNN)

Investigations were also carried out, during the present study on the suitability of probabilistic neural networks for fault identification. Probabilistic Neural Networks (PNNs) find their application mainly in classification problems (Bose and Liang, 1996). Some of the salient features of such networks are briefly described here.

A Probabilistic Neural Network bases itself on Bayes' Rule which describes the probability of the presence of a particular fault conditional to the observation of a certain symptom as

$$P(H_i / E) = \frac{P(E / H_i)\ P(H_i)}{\sum_{n=1}^{k} P(E / H_n) P(H_n)} \tag{3.30}$$

where

$P(H_i / E)$	the probability that fault H_i is present, given a symptom E
$P(E / H_i)$	probability that symptom E will be observed when fault H_i is present in absence of any specific symptom

$P\left(H_i\right)$ the apriori probability that fault H_i is present in absence of any specific symptom.

k number of faults

The following computational procedure can be followed.

$P\left(H_i\right)$ Information generally obtained from machine history, or otherwise taken as unity for all possible faults

$\sum_{n=1}^{k} P\left(E/H_n\right)P(H_n)$ This term in the denominator is constant for all i

Given a symptom E, the network computes the conditional probability $P\left(H_i/E\right)$, of the presence of faults (i = 1,2 ... k). A comparison is then made:

if $P\left(H_i/E\right) > P\left(H_j/E\right)$ for all $j \neq i$, then the fault is H_i, or

if $P\left(E/H_i\right) > P\left(E/H_j\right)$ for all $j \neq i$ then the fault is H_i.

$P\left(E/H_i\right)$ is computed using the following

$$P(E/H_i) = \frac{1}{(2\pi)^{m/2}\sigma_i^{m} n_i} \sum_{j=1}^{n_i} \left[\frac{-(E-E_j^i)^T(E-E_j^i)}{2\sigma_i^2}\right] \tag{3.31}$$

where

m number of observations e in every training symptom pattern E

n_i number of training symptom patterns pertaining to the i^{th} fault H_i

σ_i smoothing parameter computed using radial basis functions

The probabilistic neural network, in addition to the input layer, has two hidden Layers and an output layer (Fig.3.12). Its major difference with a back-propagation network is that it can be constructed after only a single pass of the training data sets. Also, the activation function is statistically derived from estimates of the probability density functions based on training patterns.

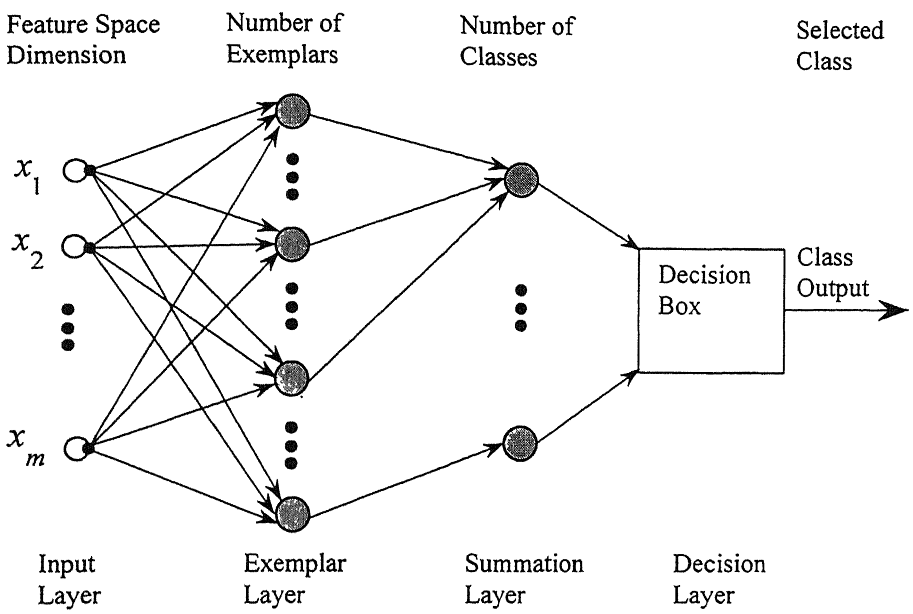


Figure 3.12 Probabilistic Neural Network

Fault simulation on the rotor rig, data acquisition and processing and network training and validation are described next in chapters 4 and 5.

CHAPTER 4

EXPERIMENTAL SET-UP AND DATA ACQUISITION

The experimental work was carried out on a commercially available rotor rig. Several faults were deliberately introduced in the rotor assembly. Vibration signals were acquired from a number of appropriate stations on the rig, through Virtual Instrumentation techniques of LabVIEW software. The vibration signals were processed and fed to the neural network simulator which was developed using MATLAB toolboxes.

4.1 Machinery Fault Simulator

The experimental rotor rig (Machinery Fault Simulator, make Spectraquest, type 2) consists of a basic framework involving a shaft supported in bearings and driven by a D.C. motor (Figs. 4.1 and 4.3). The kit also carries a reciprocating mechanism, which can be driven through a gearbox. The gearbox, in turn is driven by a belt drive consisting of two V-belts. A large number of rotating machinery faults can be introduced in this MFS and realistic dynamic conditions under rotation can be created. MFS is modular in design to facilitate easy removal/replacement of components. Additionally, the kit carries a number of components with deliberately introduced faults. This facilitates introduction of faults in the rotor assembly in a controlled manner. The present study was restricted to common faults in rotating shafts, belt drives and gearboxes. Faults in reciprocating mechanisms were not simulated. The rig has an adjustable speed range of 0 - 6000 rpm (0-100 Hz). Sweep rate adjustment in this speed range is also possible in order to simulate coast-up/coast-down operations. Different loading conditions can be simulated in MFS by setting torque in an adjustable torque magnetic brake.

4.2 Instrumentation

Location of sensors is crucial to any instrumentation scheme. In a rotating machinery, shafts are normally more flexible in transverse direction than other rotating components

and shaft deflection is considered to be an appropriate measure for rotor faults like mass unbalance, bent shaft, rotor rub etc. Bearings form the interface between rotating shafts and stationary base and faults in either of these make the bearings most susceptible to damage. Also, vibration signals from most of the components get transmitted to bearings and therefore bearing housing is an appropriate location for mounting vibration sensors. Vibrations sensed at foundation/bed-plates also provide useful information since a substantial portion of energy gets transmitted through the machine foundation. Measurement of casing vibration provides a view to the relative motion of casing with respect to the shaft and free space.

During the present study, seven sensors were used to pick up the vibration signals at the following locations (Figs. 4.2 and 4.3):

1 Shaft displacement in the vertical direction at left side bearing end: Sensor - non-contact type eddy current displacement probes (Proximitor), model 3300 RAM, Bently Nevada make ; associated transducer and TK 15 signal conditioner.

2 Shaft displacement in the horizontal direction at left side bearing end: Sensor - same configuration as in 1 above.

3 Keyphasor measurement for phase information (a small plate externally attached to the shaft was used to produce the required trigger signal): Sensor - same configuration as in 1 above.

4 Bearing housing vibration, in the vertical direction at the right side bearing: Sensor - accelerometer (model 4374, Bruel and Kjaer make) along with a charge amplifier, (model 2635, of the same make).

5 Bearing housing vibration, in the horizontal direction at the right side bearing: Sensor - same configuration as in 4 above.

6 Bearing housing vibration, in the axial direction at the right side bearing: Sensor - same configuration as in 4 above.

7 Foundation vibration at its mid-span: Sensor - same configuration as in 4 above.

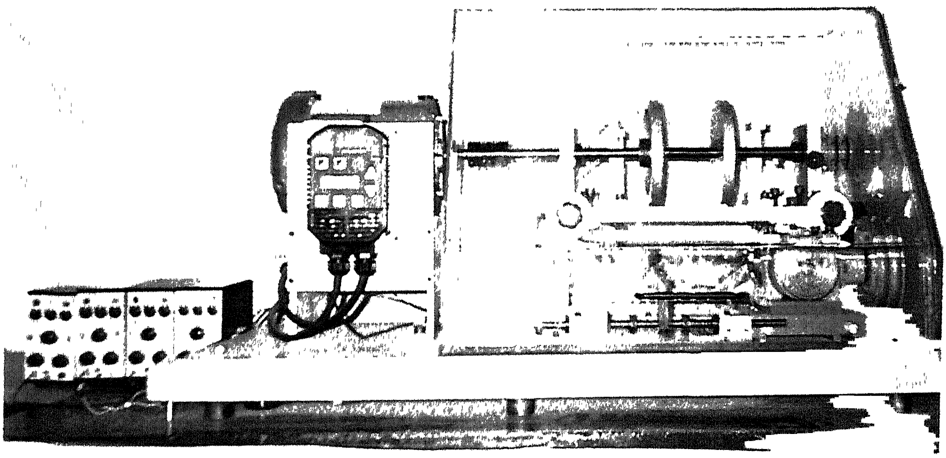


Figure 4.1 Fault Simulator and Charge Amplifiers

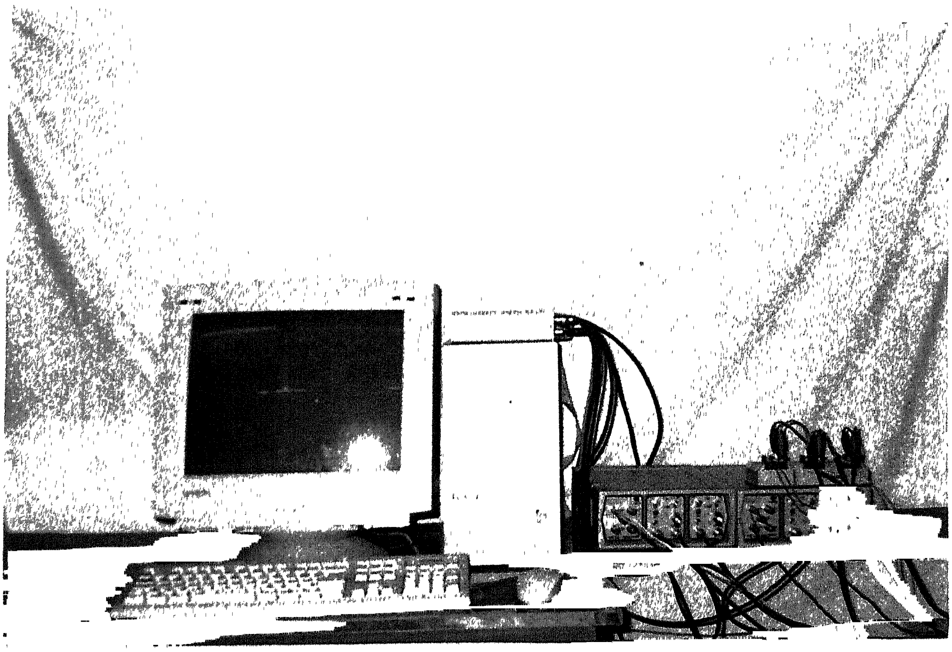


Figure 4.2 Instrumentation and Data Acquisition Devices

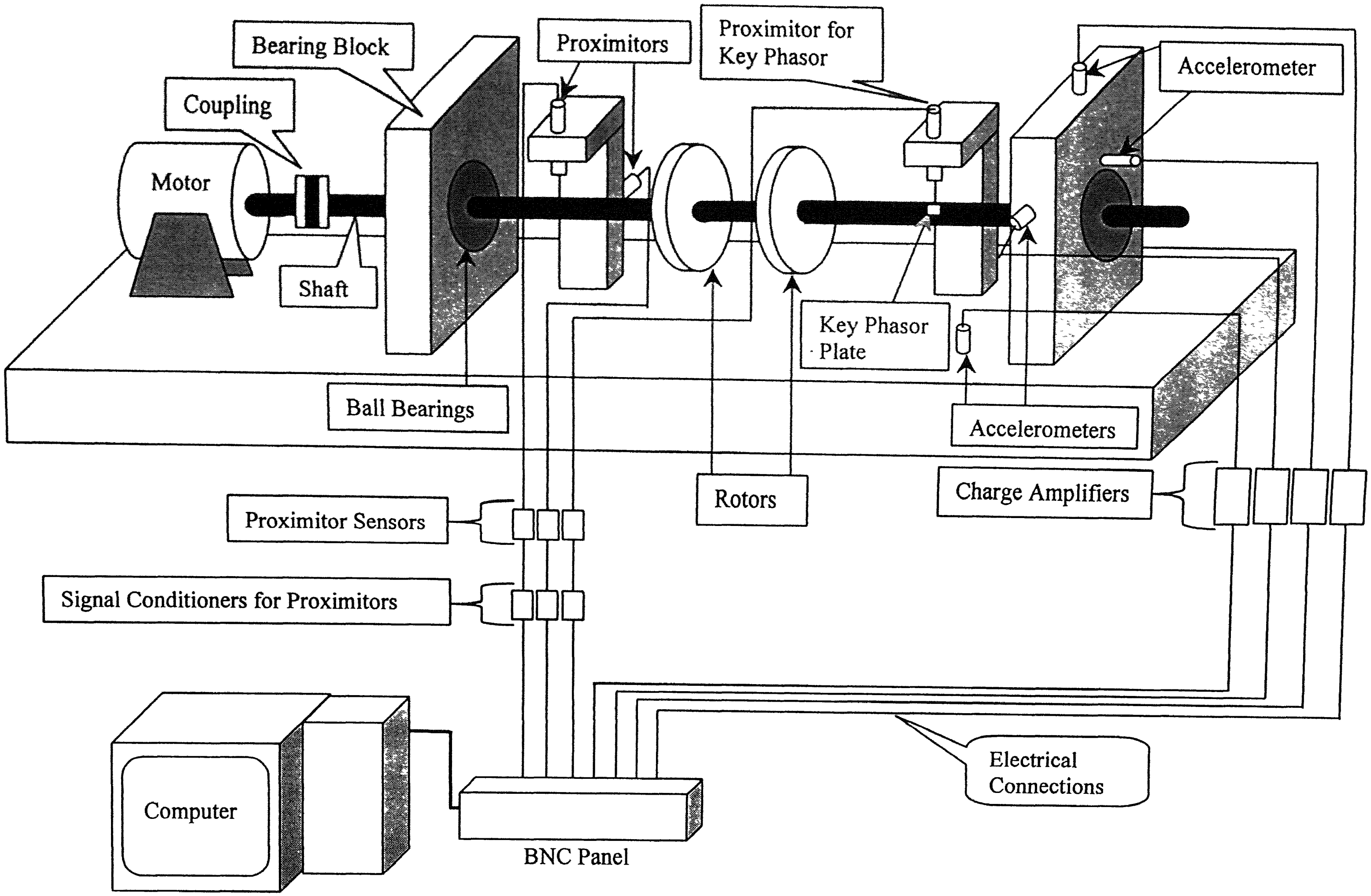


Figure 4.3 Schematic Diagram of the Experimental Set-up

Vibration data is obtained simultaneously at the above seven stations and acquired into a P-III computer through a Data Acquisition Card (AT-MIO-16E-1, National Instruments, Texas) and LabVIEW software. LabVIEW, like C or BASIC is a general-purpose programming system and includes for data acquisition, data analysis, data presentation, and storage. It uses a graphical programming language, G, to create programs in a block diagram form. LabVIEW programs are called *Virtual Instruments* (*VI*s) because of their appearance, and their operation that can imitate actual instruments. An interactive user interface of a VI, called *front panel* is used to pass on user defined parameters into the program, called *block diagram* and program's output parameters are displayed in the front panel. Since user interactive programs can be written for various applications and output parameters of the program can be displayed in text as well as graphical form, it offers tremendous potential for developing wide variety of applications. The front panel is used for data input in form of text box, knobs, buttons etc. and output in the form of text box, figure box, graphs, plots etc. Modular programming techniques make LabVIEW suitable for simulating a large number of instruments limited only by the data acquisition board used.

4.3 Data Acquisition, Storage and Display

A Virtual Instrument with the following facilities in a single Front Panel (Fig. 4.4) was developed, in LabVIEW, during the present study.

(a) Time domain vibration data acquisition from seven channels with following user controls (figures/text within brackets indicate default values):

- (i) Scan Rate (1000).
- (ii) Number of data points to be read before display (4096).
- (iii) Device and Channel numbers from which to acquire data (Device no. 1 and Channel nos. 0 to 6).
- (iv) Whether to use trigger for start of data acquisition (No).
- (v) Channel to be used for trigger signal (0).
- (vi) Trigger to be used on rising/falling signal levels (none).
- (vii) Trigger level (0).

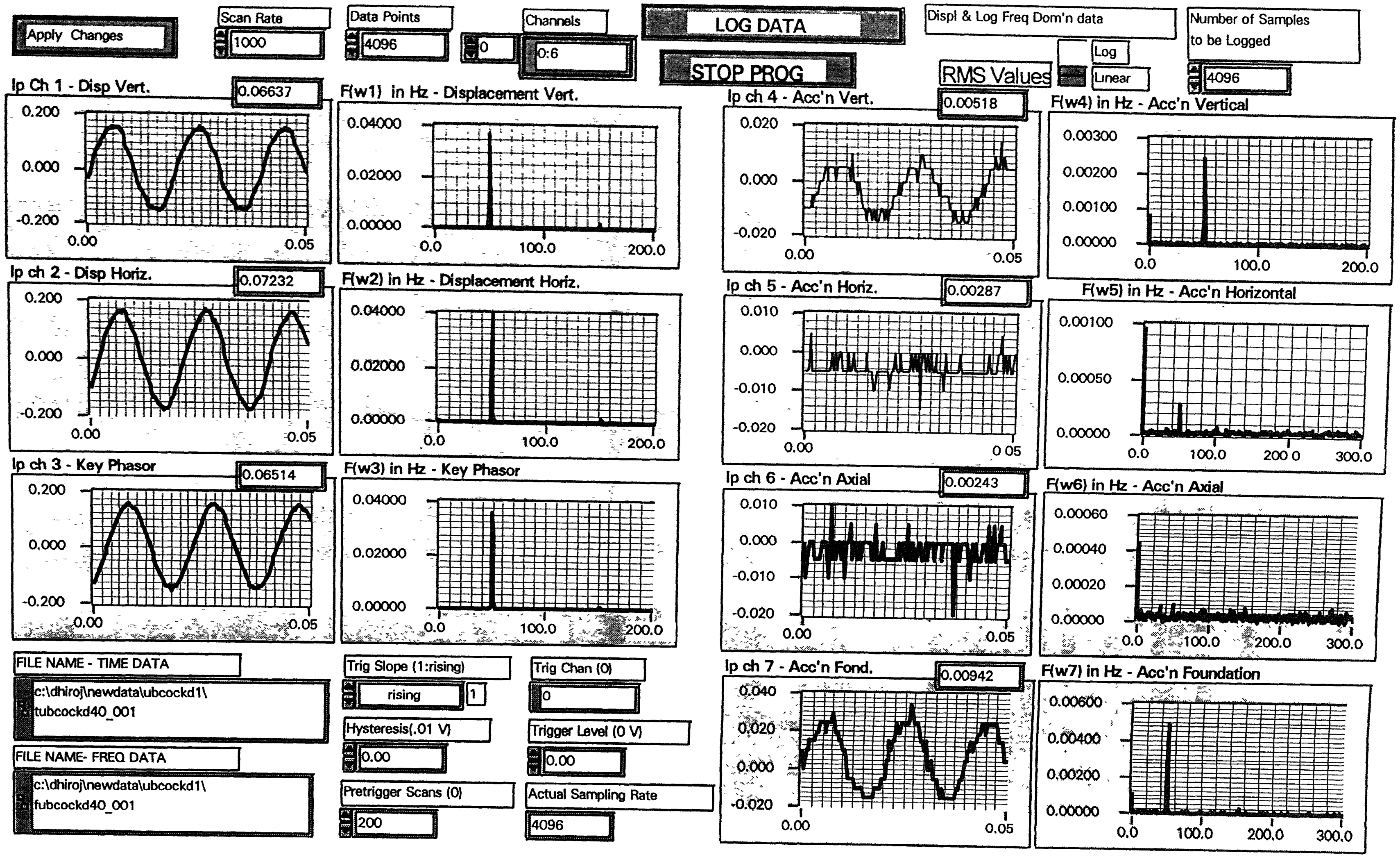


Figure 4.4 Typical Front Panel of Data Acquisition, Display and Logging VI

(viii) Trigger hysteresis (0).

(ix) Number of data points to be recorded before trigger (200)

(b) Conversion of time domain data to Frequency domain data by Fast Fourier Transform (FFT) after filtering with a Hanning window with user control for number of data points to be used for the transformation (4096).

(c) Root Mean Square (RMS) value evaluation of time domain data for each channel.

(d) Option for data logging of both time and Frequency domain data of all seven channels in user defined file names in spreadsheet format.

(e) The display features include the following

(i) Graphical Time domain data.

(ii) Graphical Frequency domain data with control for linear/log scales.

(iii) RMS values of time domain data.

(iv) Actual sampling rate.

4.4 Fault Simulation

A preliminary test was done to check the working of the Machinery Fault Simulator. A large number of faults were simulated on and vibration signals were obtained to make a visual inspection of the specific features in the corresponding Fourier Spectrum. However, for the purpose of development of the neural network, the number of considered faults was restricted to 6, in order to keep data handling at a manageable level. The eighteen faults simulated for the preliminary check are:

1. No Fault – same as factory condition of MFS
2. 5.2 gm mass unbalance on right rotor
3. 5.2 gm mass unbalance on left rotor
4. 5 mil parallel misalignment in horizontal plane
5. 3 mil angular misalignment in horizontal plane
6. Combination of faults at 2 and 4 above
7. Combination of faults at 2 and 5 above
8. Damaged inner race of right bearing
9. Damaged outer race of right bearing
10. Ball Spin in right bearing

11. Cocked left rotor
12. Mechanical rub on shaft near left bearing
13. No Fault with belt drive and gear box connected
14. Tight Belt with belt drive and gear box connected
15. Loose Belt with belt drive and gear box connected
16. As in 13 above and mass unbalance in gearbox output shaft
17. As in 14 above and mass unbalance in gearbox output shaft
18. As in 15 above and mass unbalance in gearbox output shaft

Faults (1) to (12) were simulated without the belt drive i.e. gearbox and reciprocating mechanisms, while the rest were simulated without the reciprocating mechanism alone. Vibration data was acquired and stored into the computer for each of these faults. Fourier Spectrum of typical vibration signals in the vertical direction picked up by the accelerometer at the bearing housing for each of the eighteen faults are displayed in Figs. 4.5 (a-r). It can be seen that each fault carries its own distinct signature in the frequency domain, and the location and magnitude of the peaks can serve as a useful tool in the reverse problem of identification of an unknown fault from a frequency domain signal.

The following six cases were chosen, from the 18 faults listed earlier, for development of the Neural Network Simulator.

1. No Fault – same as factory condition of MFS
2. 5.2 gm mass unbalance on right rotor
3. Damaged inner race of right bearing
4. Damaged outer race of right bearing
5. Ball Spin in right bearing
6. Cocked left rotor

A total of one hundred and ten sets of vibration signals were acquired for each of the above six cases. Each set includes signals from the seven sensors mentioned earlier. Out of the six hundred and sixty signals, six hundred were used for neural network training and the remaining sixty were set aside for testing the trained algorithm. Time domain vibration signals were acquired at a scan rate of 2560 samples/sec. Frequency domain transformation was carried out on line using an FFT algorithm.

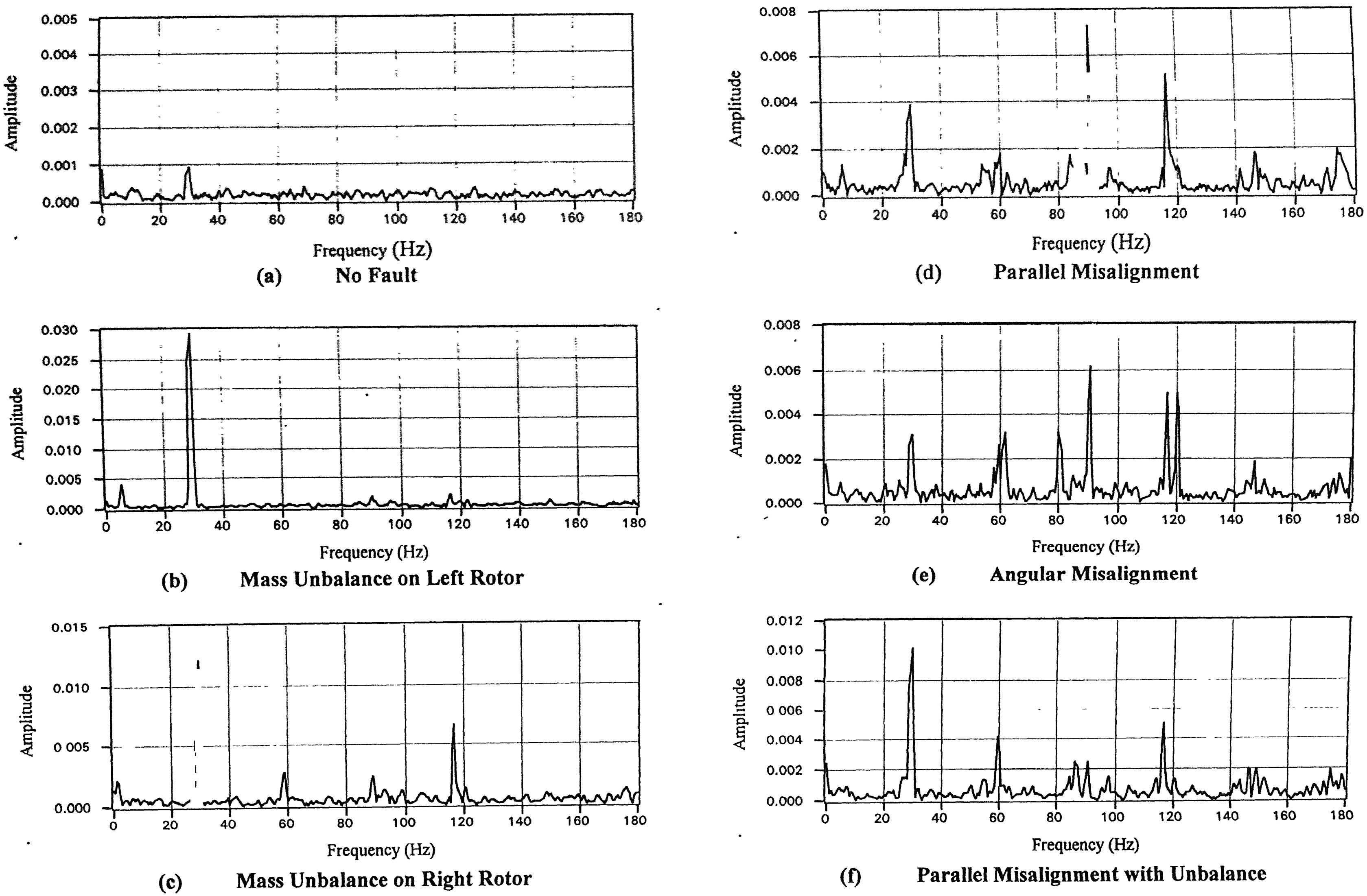
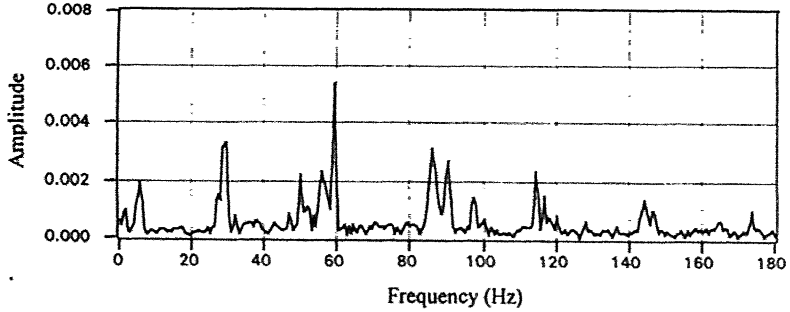
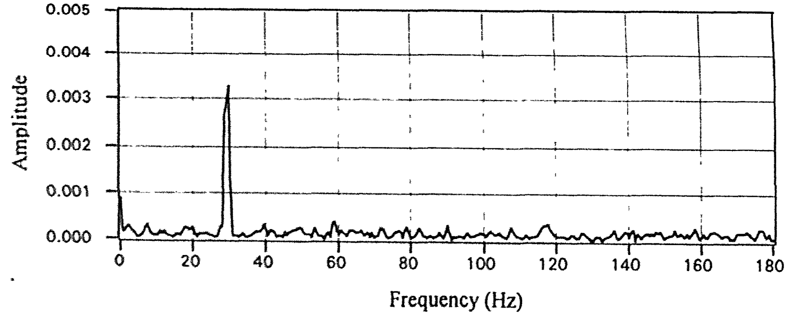


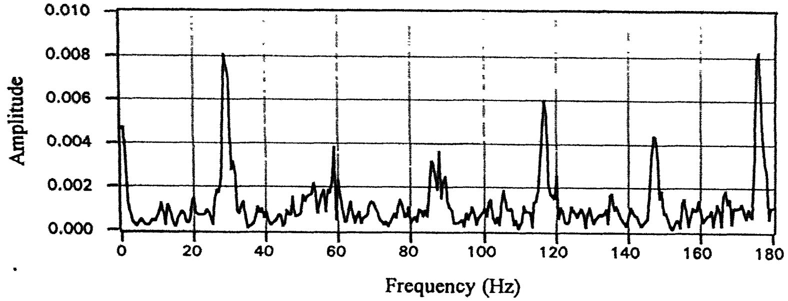
Figure 4.5 Bearing Housing Acceleration (Vertical Direction): Rotor Speed 1800 rpm



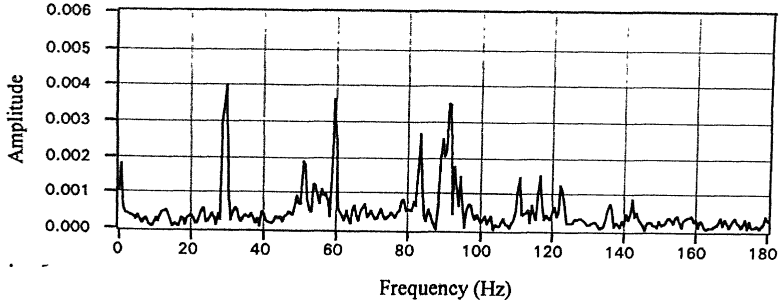
(g) Parallel Misalignment with Unbalance



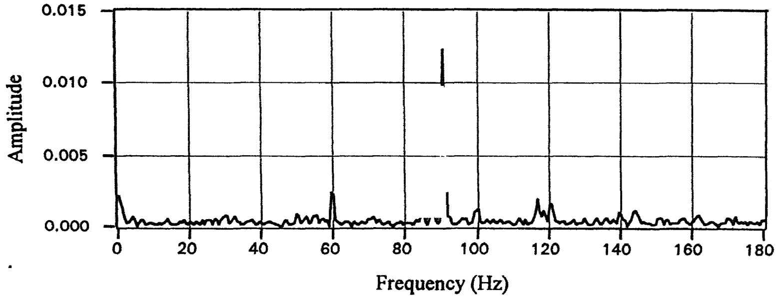
(j) Ball Spin in Bearing



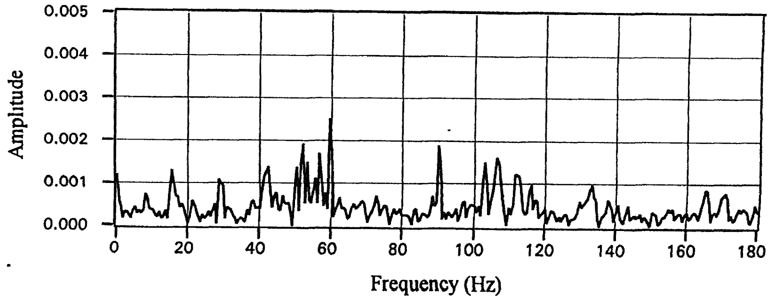
(h) Damaged Bearing Inner Race Defect



(k) Cocked Rotor

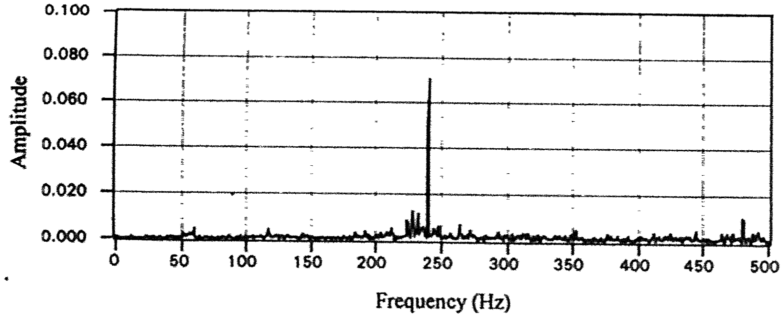


(i) Damaged Bearing Outer Race

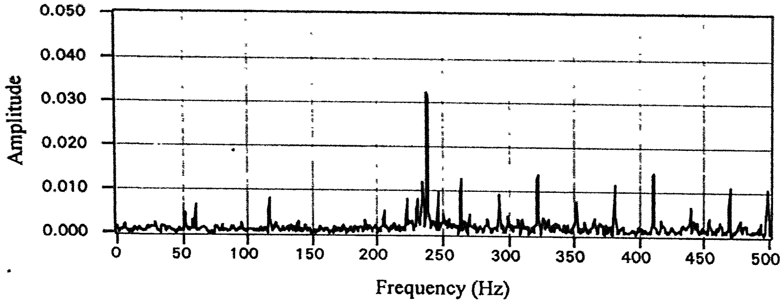


(l) Shaft Rub

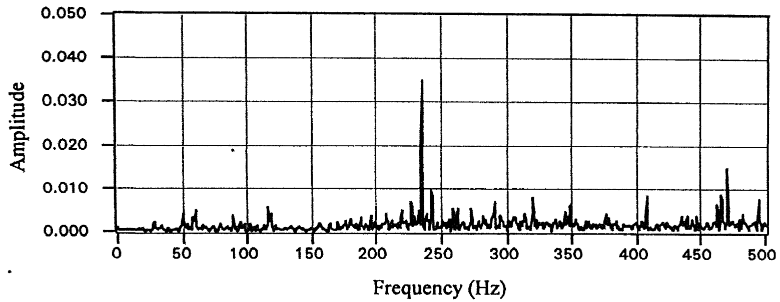
Figure 4.5(Contd.) Bearing Housing Acceleration (Vertical Direction): Rotor Speed 1800 rpm



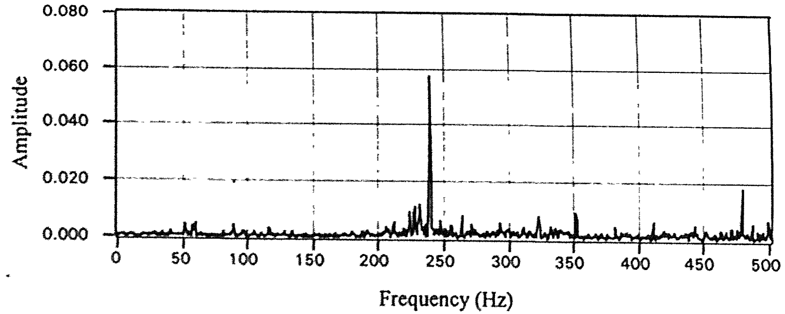
(m) No Fault with Gearbox Running



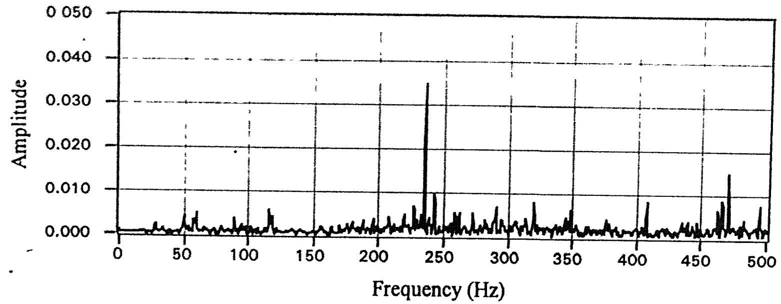
(n) Tight Belt



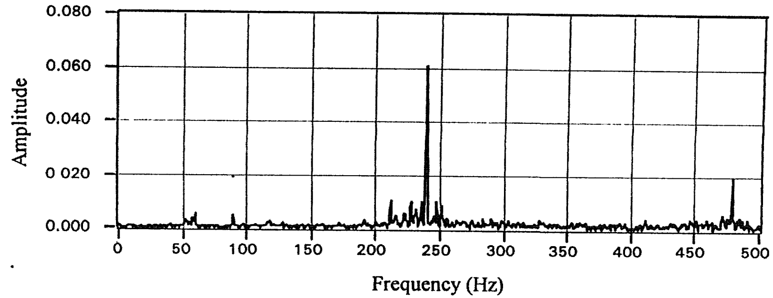
(o) Loose Belt



(p) Unbalance on Gearbox Driven Shaft



(q) Tight Belt Combined with Fault (p) above



(r) Loose Belt Combined with Fault (p) above

Figure 4.5(Contd.) Bearing Housing Acceleration (Vertical Direction): Rotor Speed 1800 rpm

CHAPTER 5

NETWORK TRAINING AND VALIDATION

Input to Artificial Neural Networks is normally in matrix form with a column representing a set of data. Elements in this column are indicative of the salient features (expressed in numeric form) of a data set. Extraction of the significant features from raw data is crucial to any condition monitoring exercise. Extraction of these features from a vibration signal can be arduous since they are almost always contaminated with noise. However, in the present exercise since the rotor rig is custom made for fault simulation, the noise level is low and the signals did not require any treatment for noise removal.

5.1 Feature Extraction

Feature extraction from vibration signatures to characterize particular machine faults has received considerable attention from researchers. Elkordy, Chang and Lee (1994) used state variables computed from the strain mode shapes and displacement mode shapes for structural damage diagnosis and condition monitoring of multi-storied buildings. Canonical correlation analysis was used by Mayes (1994) to study oil-whirl problem and reactive load dependent behavior of power generator. McCormick and Nandi (1997) extracted time-invariant features from stationary time series by estimating zero-lag moments. However, these procedures consume more computational time and memory since the moments and cross-moments are required to be calculated. FFT is widely recognised as a powerful tool to characterize most of the common faults in rotating machinery. Frequency spectrum is most commonly used and understood by most of the vibration professionals. Also, frequency domain data are easy to obtain from time series since a number of FFT analyzers, both digital and analogue, are available off the shelf. However, randomness in the frequency spectrum along with shifting frequencies does not allow simple diagnosis. In the present work an algorithm has been developed for automatic detection of peaks in a Fourier Spectrum. The algorithm can take the frequency

domain spectrum as input and provide the amplitudes and frequency locations of the prominent peaks, as output.

The peak detection algorithm addresses the following aspects.

(a) A frequency spectrum would normally comprise of a large number of crests and troughs. While the visual process of peak identification is subjective (dependent on the judgement of the viewer), a computer algorithm will require quantification and comparison in order to decide which of the crests on the spectrum can be labeled as 'peaks'. One way of quantification would be to first compute the mean of the signal and then setting a threshold, which can be a function of this mean. The next step would be comparison of the amplitude at every frequency on the spectrum with this threshold. If the amplitude is higher than the threshold then it can be said that a peak exists at the particular frequency. The algorithm developed in this study follows the above logic. Also, flexibility is provided to define the threshold as some multiple of the computed mean, in a user-interactive manner.

(b) Searching for peaks is carried out over small frequency bands rather than discrete individual frequencies on the spectrum. This is done in order to account for the minor shifts in the amplitudes due to a variety of reasons, including windowing.

(c) A frequency spectrum always carries a large number of peaks. Some may be located at natural frequencies of the machine, while others may occur at harmonics of the rotational speed. Search at all possible frequencies on the spectrum may be time consuming. A decision should be made regarding the specific frequencies at which the algorithm should conduct a search.

(d) Neural networks work on the principle of multiplying various synaptic weights to an input data to generate the input to its activation function. The importance of all the peaks in this scheme cannot be identical and therefore the peak-detection algorithm should retain information about the relative magnitudes of the peaks.

Automated peak detection was carried out, during the present study, for the first five harmonics of the rotational speed. The program was written in MATLAB, which utilises

the LabVIEW data in a spreadsheet form. The Flow-Chart of the algorithm is given in Fig. 5.1.

It has user interactive control for the following values:

(a) Rotational Speed of the machine

(b) Upper level of harmonic up to which harmonics are to be evaluated

(c) Frequency band around the frequency of interest for determining presence/ absence of peaks

(d) Whether half harmonics/ fractional harmonics are also to be evaluated

(e) Multiple of average amplitude to be taken as a threshold for defining a peak

(f) Type of data output required. Data output can be in five different types as listed in the algorithm in Fig. 5.1. This facilitates use of various types of peak detection data for neural networks or other applications. For this study, ratio of mean amplitudes of frequency bands around the peak to the mean amplitude of particular channels was used.

Figs. 5.2 - 5.7 show typical sets of the actual Fourier Spectra and plots of peaks identified through algorithm discussed above. The plots pertain to vertical and horizontal acceleration measured at the bearing housing for each of the six faults under consideration.

During this study, while measurements were taken with seven sensors, which included both proximity pick-ups and accelerometers, it was found that shaft displacement signals acquired by proximity probes do not provide any additional information for the present rotor set up. Also, phase information from the keyphasor was not utilised during this study. Therefore, further data processing was carried out only for the four accelerometer channels, namely (i) bearing housing vertical acceleration (ii) bearing housing horizontal acceleration (iii) bearing housing axial acceleration (iv) foundation vibration. This reduction helps in reducing the computational effort considerably.

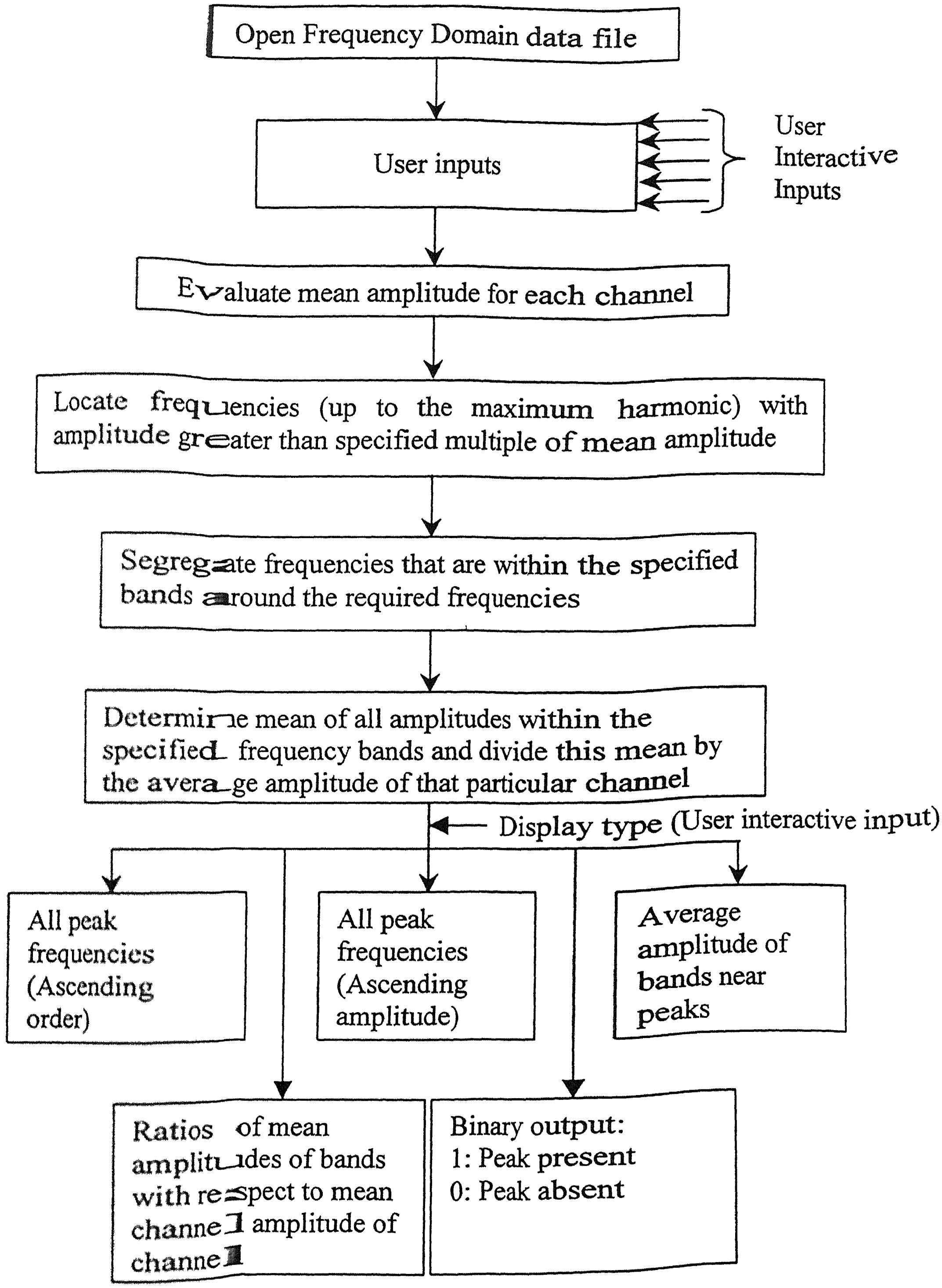
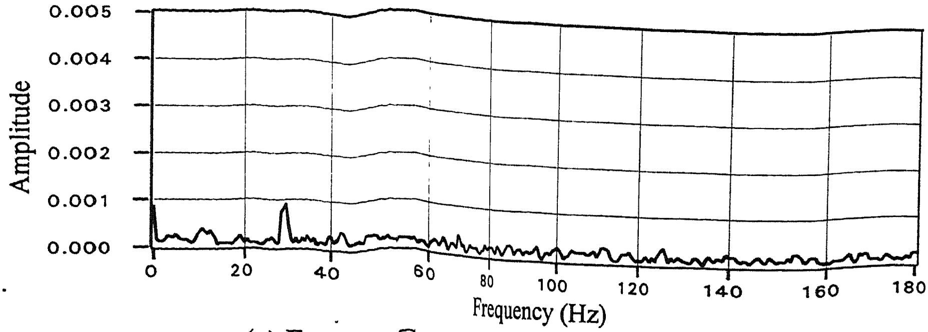
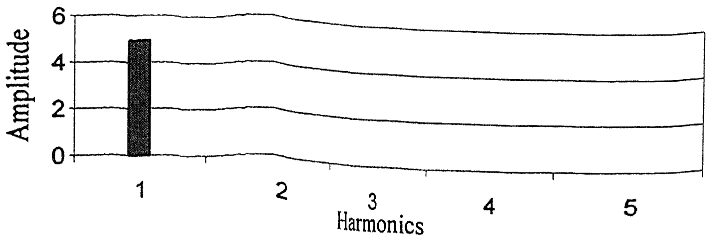


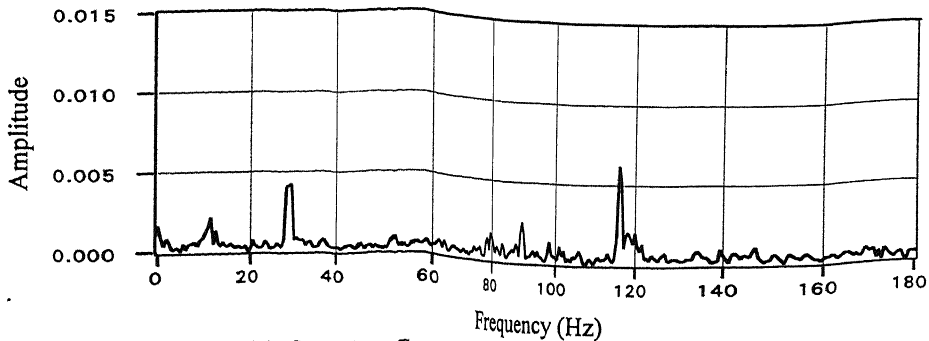
Fig. 5.1 Algorithm for Peak Detection from Frequency Domain Data



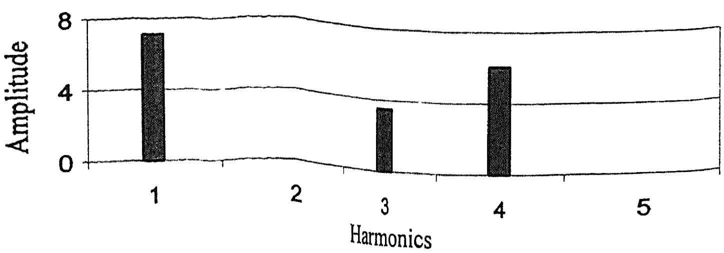
(a) Fourier Spectrum: Acceleration - Vertical



(b) Detected Peaks: Acceleration - Vertical

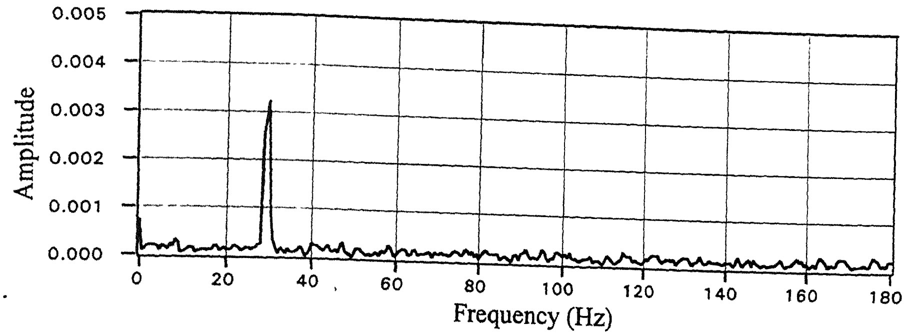


(c) Fourier Spectrum: Acceleration - Horizontal

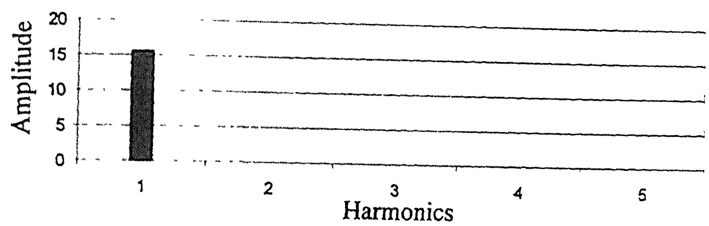


(d) Detected Peaks: Acceleration - Horizontal

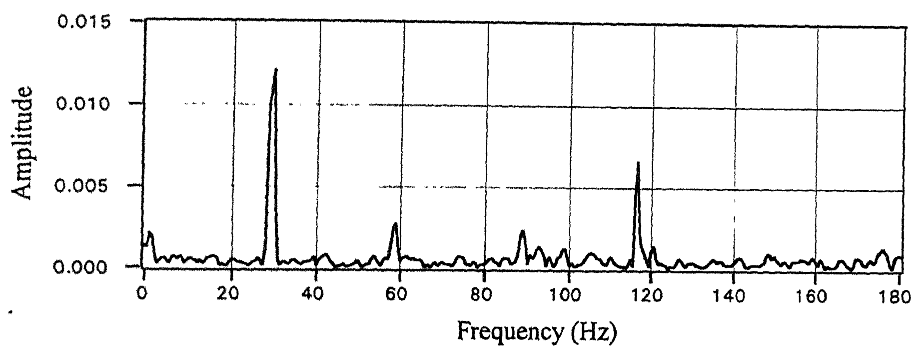
Figure 5.2 Feature Extraction: No Fault (30 Hz, Bearing Housing Acceleration)



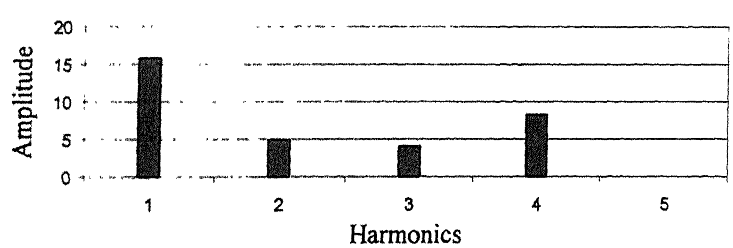
(a) Fourier Spectrum: Acceleration - Vertical



(b) Detected Peaks: Acceleration - Vertical

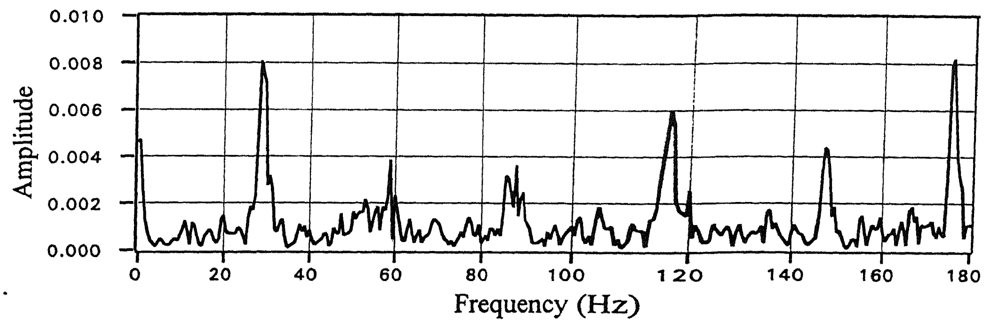


(c) Fourier Spectrum: Acceleration - Horizontal

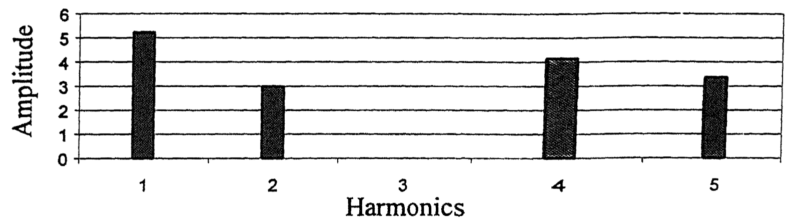


(d) Detected Peaks: Acceleration - Horizontal

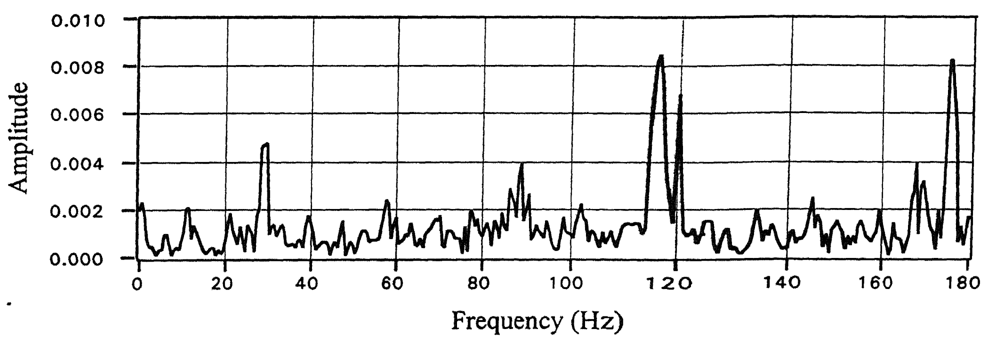
Figure 5.3 Feature Extraction: Mass Unbalance (30 Hz, Bearing Housing Acceleration)



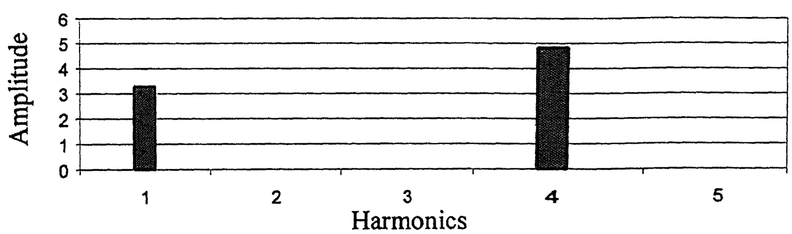
(a) Fourier Spectrum: Acceleration - Vertical



(b) Detected Peaks: Acceleration - Vertical

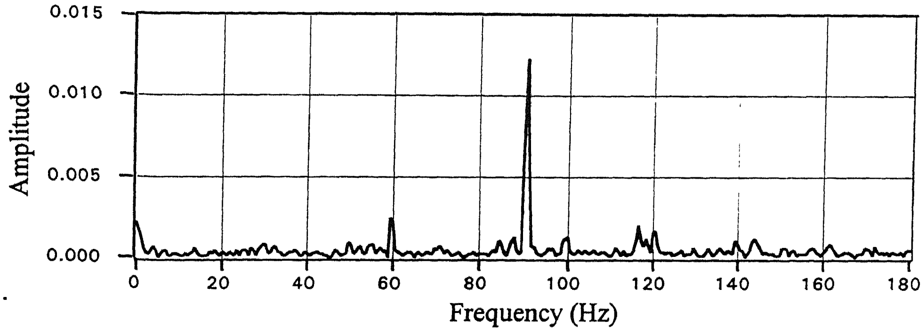


(c) Fourier Spectrum: Acceleration - Horizontal

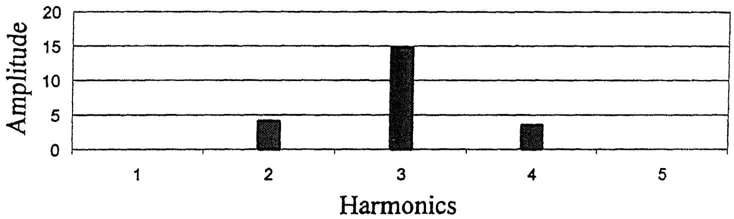


(d) Detected Peaks: Acceleration - Horizontal

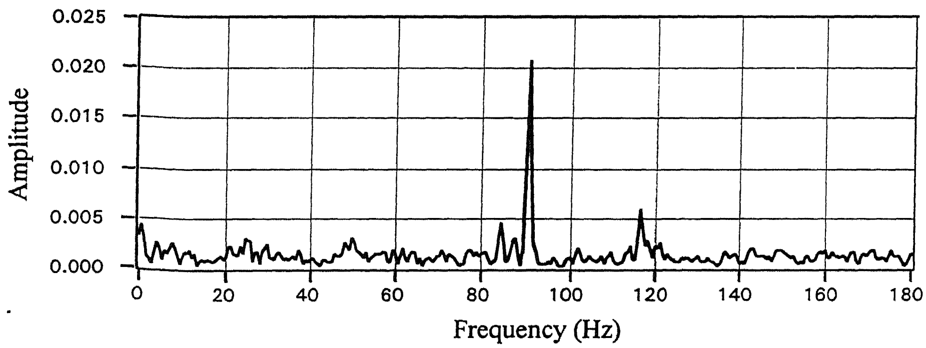
Figure 5.4 Feature Extraction: Damaged Bearing Inner Race (30 Hz, Bearing Housing Acceleration)



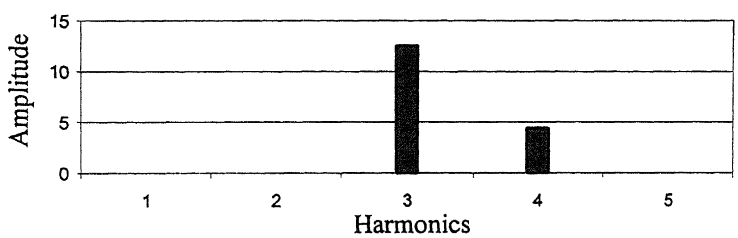
(a) Fourier Spectrum: Acceleration - Vertical



(b) Detected Peaks: Acceleration - Vertical

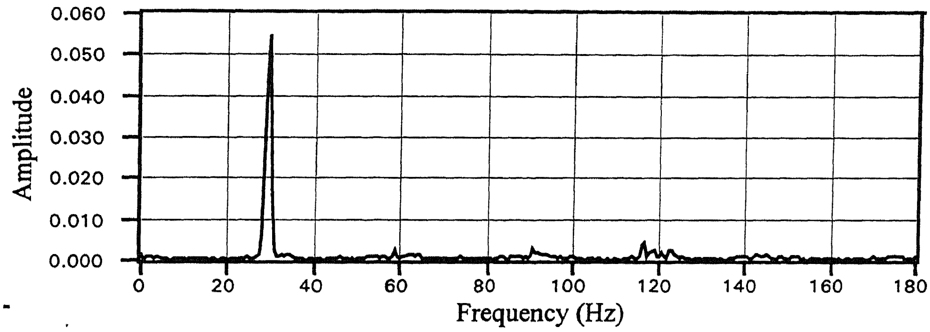


(c) Fourier Spectrum: Acceleration - Horizontal

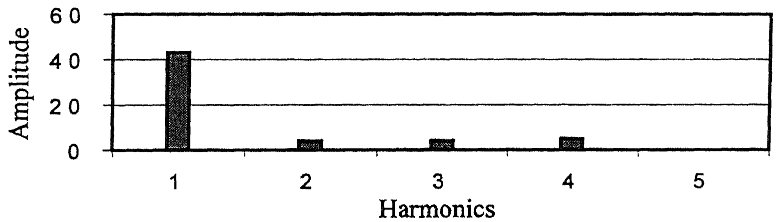


(d) Detected Peaks: Acceleration - Horizontal

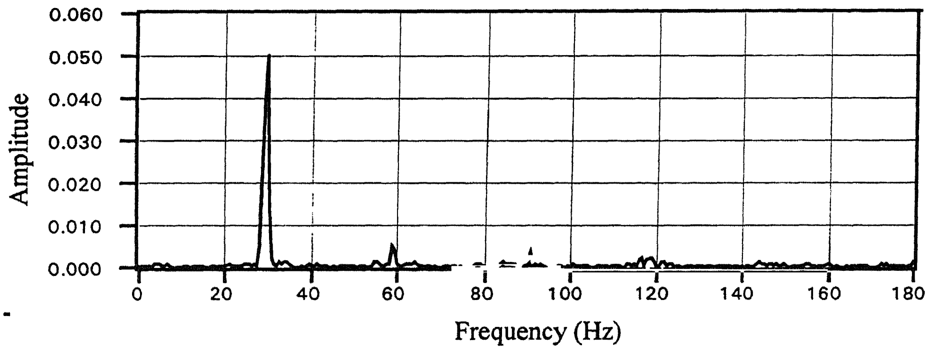
Figure 5.5 Feature Extraction: Damaged Bearing Outer Race (30 Hz, Bearing Housing Acceleration)



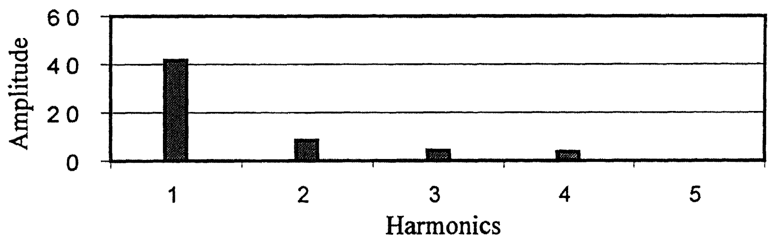
(a) Fourier Spectrum: Acceleration - Vertical



(b) Detected Peaks: Acceleration - Vertical

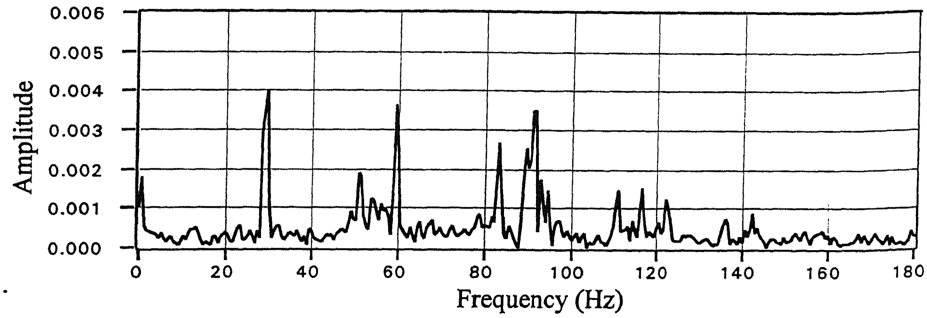


(c) Fourier Spectrum: Acceleration - Horizontal

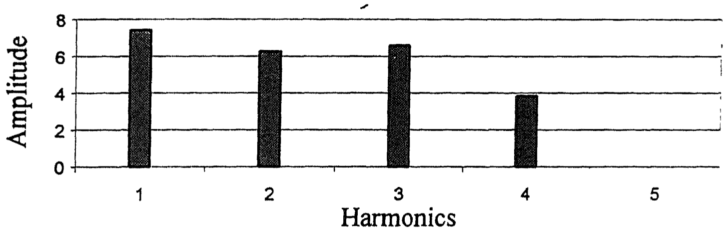


(d) Detected Peaks: Acceleration - Horizontal

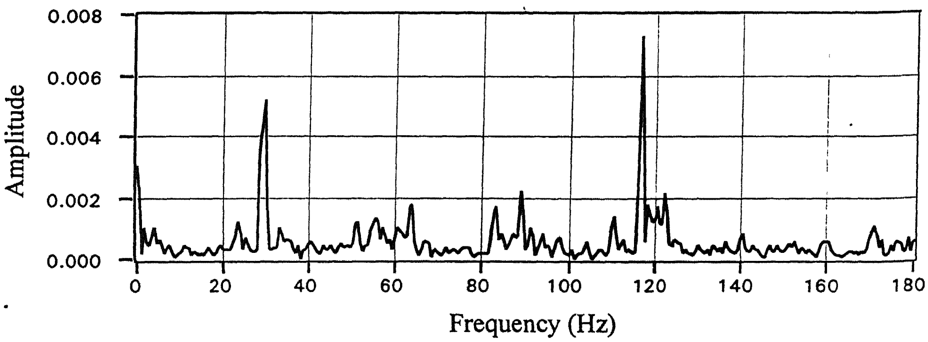
Figure 5.6 **Feature Extraction: Bearing Ball Spin (30 Hz, Bearing Housing Acceleration)**



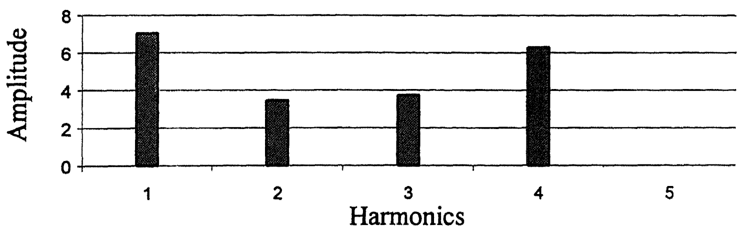
(a) Fourier Spectrum: Acceleration - Vertical



(b) Detected Peaks: Acceleration - Vertical



(c) Fourier Spectrum: Acceleration - Horizontal



(d) Detected Peaks: Acceleration - Horizontal

Figure 5.7 Feature Extraction: Cocked Rotor (30 Hz, Bearing Housing Acceleration)

5.2 Training Vector

The plots in Figs.5.2(b, d), 5.3(b, d)...5.7(b, d), provide the training vector for the neural network. For example, the peaks identified in Fig.5.3(b), which pertain to the vertical acceleration at the bearing housing for the mass unbalance fault, can be listed as

Harmonic Number	Peak Amplitude
I	15.548
II	0
III	0
IV	0
V	0

Similar lists can be obtained from the other three sensors. These lists can be combined to form Table 5.1.

The pattern in column three of Table 5.1 forms the input vector (of size 20) to the neural network. The network can be trained to identify this pattern with a binary pattern of the type [0 1 0 0 0 0]. This binary pattern forms the target vector associated with mass unbalance fault.

Table 5.2 gives typical training vectors for the six fault configurations under consideration, along with the target vectors. It is to be noted that while the target vector for a particular fault remains fixed, the training vector vary depending on the variations in the individual signals. These variations can be noted from the data given in Table 5.3, which gives ten training vectors for the cocked rotor fault. The variation in training vectors is highlighted in Fig. 5.8(a-c).

Table 5.1 Typical Training Vector

Sensor Number	Harmon[redacted]ic Number	Peak Amplitude
1. Acceleration Vertical	1	15.548
	2	0
	3	0
	4	0
	5	0
2. Acceleration Horizontal	1	15.873
	2	4.942
	3	4.057
	4	8.235
	5	0
3. Acceleration Axial	1	6.479
	2	4.545
	3	3.015
	4	0
	5	0
4. Acceleration Foundation	1	13.591
	2	3.682
	3	0
	4	0
	5	0

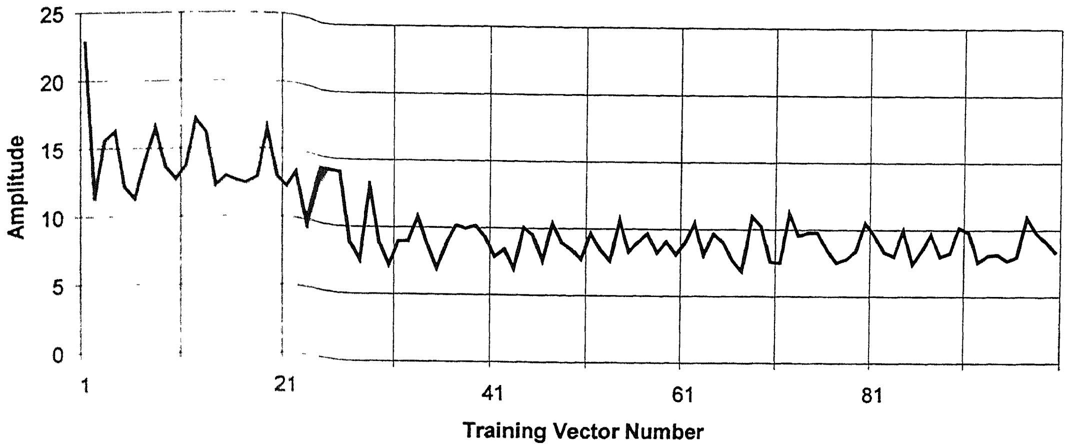
Table 5.2 Sample Training and Target Vector Sets

Sl. No.	Training Vectors						Target Vectors											
							For Backpropagation Network						For Probabilistic Neural Network					
	NF	UB	IR	BS	OR	CK	NF	UB	IR	BS	OR	CK	NF	UB	IR	BS	OR	CK
1	3.187	11.331	19.001	0	0	0	1	0	0	0	0	0	1	2	3	4	5	6
2	0	3.24	3.346	4.272	0	5.512	0	1	0	0	0	0						
3	7.146	12.094	7.561	3.654	6.26	8.183	0	0	1	0	0	0						
4	0	0	5.945	3.03	0	0	0	0	0	1	0	0						
5	0	0	4.381	0	0	3.226	0	0	0	0	1	0						
6	0	9.652	34.141	3.835	4.314	22.816	0	0	0	0	0	1						
7	0	6.081	8.464	3.212	3.588	7.048												
8	0	16.695	14.12	5.192	4.202	9.962												
9	0	5.052	5.28	5.093	4.629	0												
10	0	0	3.06	4.299	0	4.408												
11	0	0	0	0	0	18.539												
12	0	8.214	4.34	0	0	5.764												
13	0	8.632	4.631	0	7.338	14.789												
14	0	6.098	16.833	0	0	0												
15	0	3.294	0	3.391	0	0												
16	0	0	0	0	0	4.337												
17	0	3.718	0	0	0	0												
18	0	3.497	3.969	3.026	0	0												
19	0	0	0	0	0	0												
20	0	0	0	3.096	0	3.096												

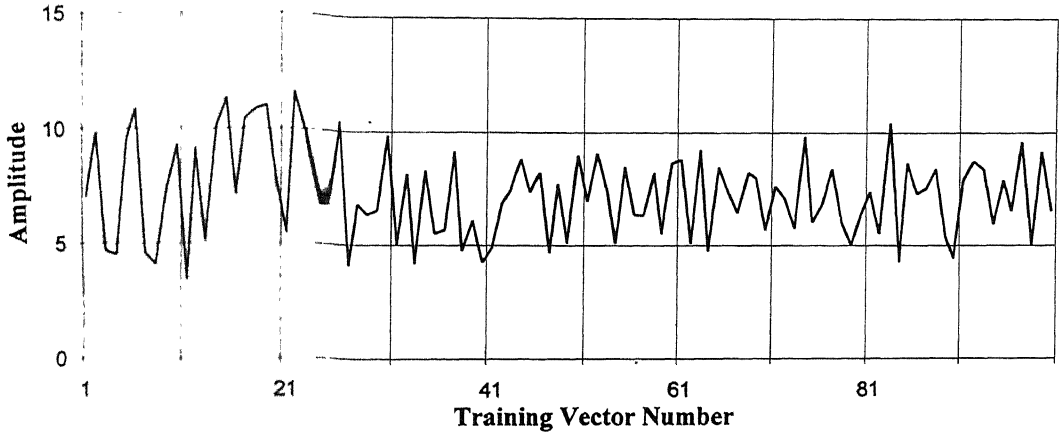
Note: NF - No Fault; UB - Mass Unbalance; IR - Damaged Bearing Inner Race;
BS - Ball Spin in Bearing; OR - Damaged Bearing Inner Race; CK- Cocked Rotor

Table 5.3 Typical Training Vectors (for Cocked Rotor Fault)

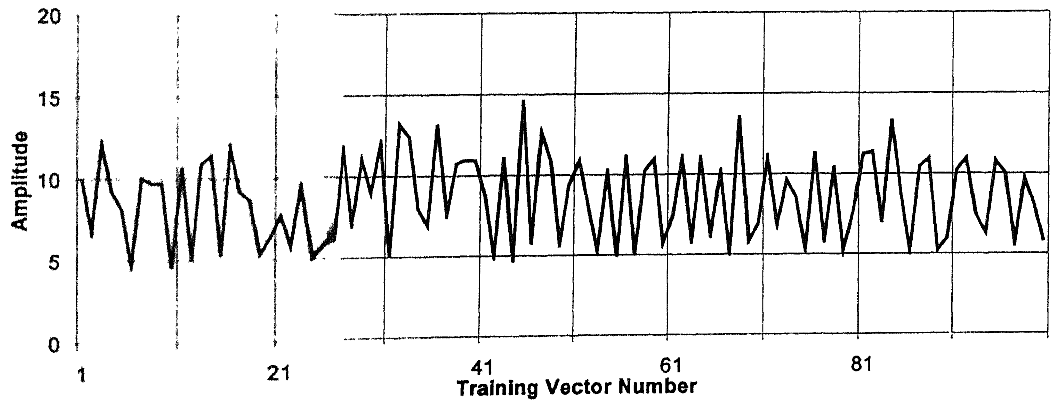
		Vector Numbers (only first ten vectors shown)									
AccelerometerDirection	Harmonics	1	2	3	4	5	6	7	8	9	10
Vertical	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	2	5.512	6.482	4.741	4.730	4.449	5.811	5.137	5.040	4.811	5.971
	3	8.183	9.215	8.716	9.104	8.911	9.628	8.961	8.276	8.402	9.565
	4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	3.226	4.176	0.000	0.000	4.147	3.209	0.000	0.000	3.344	3.069
Horizontal	1	22.816	11.333	15.603	16.274	12.157	11.289	14.104	16.537	13.626	12.727
	2	7.048	9.791	4.700	4.550	9.545	10.798	4.613	4.110	7.268	9.313
	3	9.962	6.718	12.185	9.168	8.091	4.665	10.040	9.707	9.759	4.568
	4	0.000	3.697	0.000	0.000	3.760	0.000	3.750	3.738	3.141	3.245
	5	4.408	0.000	3.955	3.468	4.654	0.000	3.977	3.545	4.339	3.128
Axial	1	18.539	19.382	18.995	15.399	18.280	20.358	18.187	18.480	15.286	19.662
	2	5.764	5.305	5.711	6.780	5.528	4.965	6.446	5.965	6.103	6.203
	3	14.789	10.674	10.815	7.737	14.075	8.279	13.141	11.986	12.566	8.513
	4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Foundation	1	4.337	3.315	3.603	4.011	4.329	3.840	3.837	3.567	3.239	3.989
	2	0.000	0.000	5.901	0.000	0.000	0.000	4.068	3.116	3.181	0.000
	3	0.000	4.027	3.008	3.454	4.317	4.783	0.000	0.000	0.000	4.851
	4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	3.096	4.011	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000



(a) First Harmonics: Horizontal Acceleration



(b) Second Harmonics: Horizontal Acceleration



(c) Third Harmonics: Horizontal Acceleration

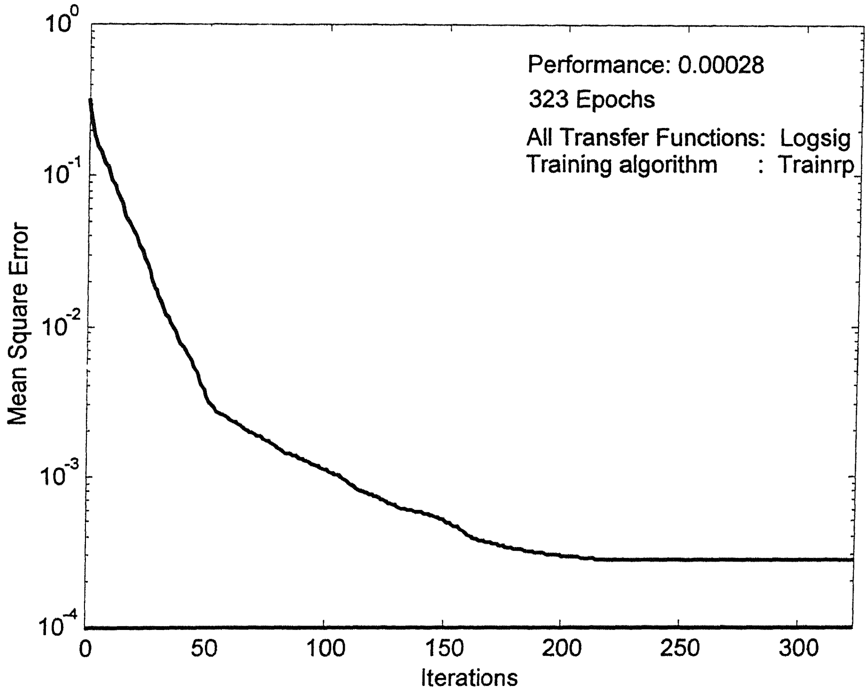
Figure 5.8 **Variation in Training Vectors for Three Harmonics of a Single Channel for Same Fault (Cocked Rotor)**

5.3 Back-Propagation Network Training

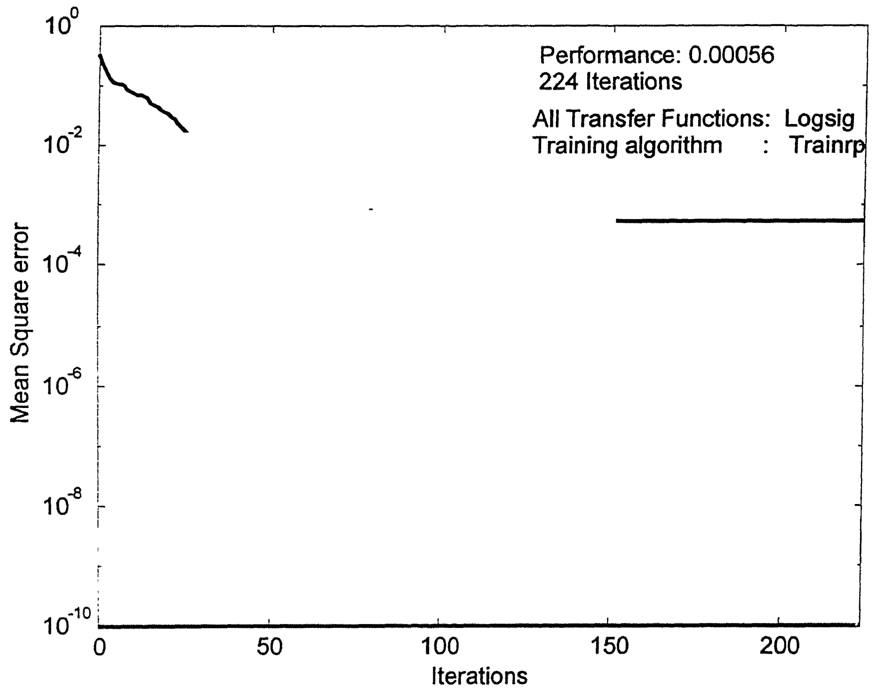
Various types of back-propagation architectures were investigated for their efficiency during training and in prediction of faults. These networks were developed using the Neural Network toolbox in MATLAB. In all these architectures the input and output vector sizes were kept constant at twenty and six, respectively. The number of hidden layers and their sizes were varied. The following architectures were tried (nomenclature 20 -h*1*-h*2*-h*3*....- h*k*- 6 etc., where h*k* denotes the number of neurons in the *k*th hidden layer):

(i) 20-8-6
(ii) 20-10-6
(iii) 20-12-6
(iv) 20-17-6
(v) 20-15-9-6
(vi) 20-17-8-6
(vii) 20-20-8-6

Architectures (i)-(iv) involve one hidden layer, while (v) and (vii) involve two hidden layers. Training was carried out for each of the above architectures using the same set of 600 training vectors. Figs. 5.9 (a-g) show the convergence patterns of these networks (Fig. 5.9(h) refers to convergence of network (vii) above using TANSIG transfer function). The convergence results have also been summarised in Table 5.4. The minimum achievable mean square error in architecture (i), 20-8-6, is 2.7778e-4. This is achieved after 323 iterations. Increase in the number of neurons in the hidden layer initially results in better performance (architectures (ii), 20-10-6 and (iii), 20-12-6), whereby lesser value of the error is achieved, through lesser number of iterations. Increase in the number of neurons in the hidden layer to 17, results in over-training and yields increased error and slower convergence. It can be seen that error can been further reduced by increasing the number of hidden layers (architectures (v)-(vii)). The minimum

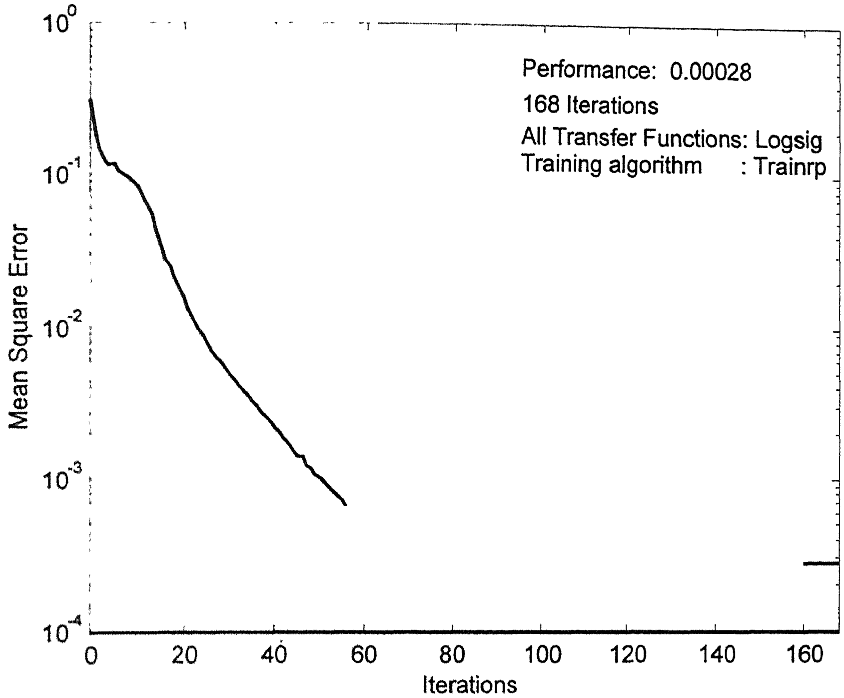


(a) Architecture 20-8- 6

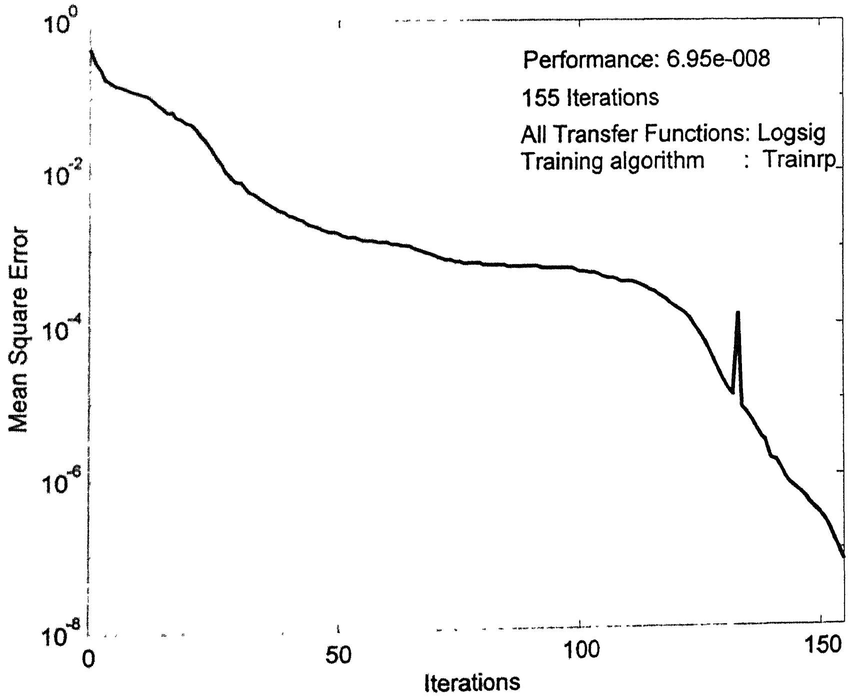


(b) Architecture 20-10-6

Figure 5.9 Convergence Pattern of Various Back-propagation Network Architectures

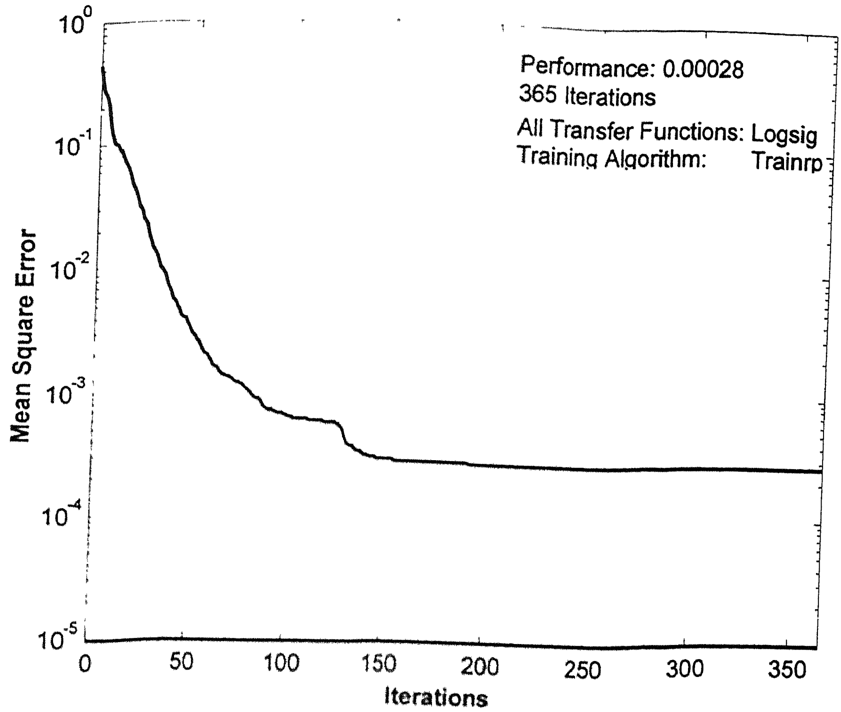


(c) Architecture 20-12-6

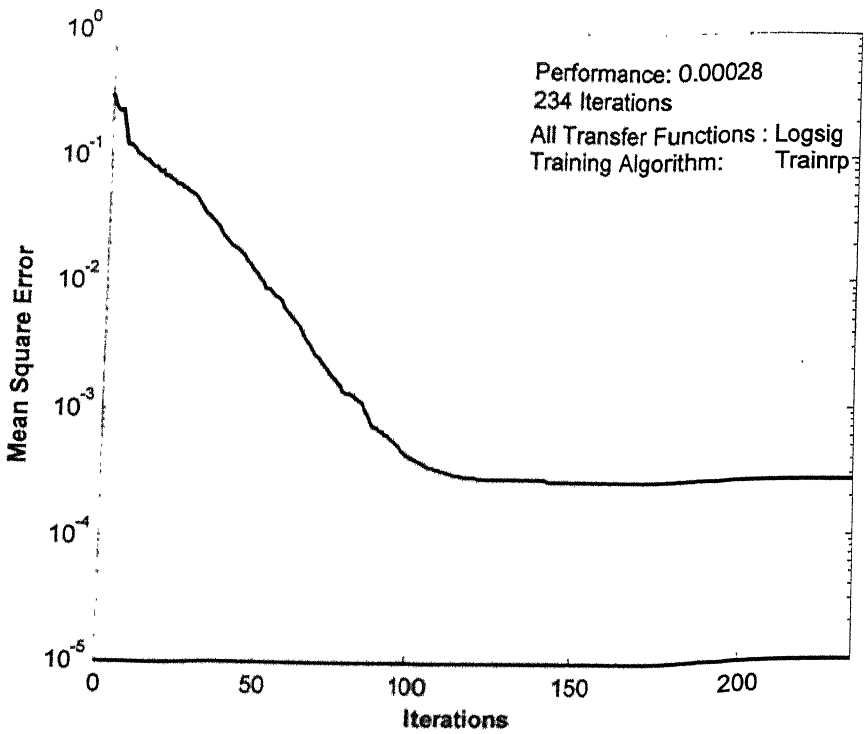


(d) Architecture 20-15-9-6

Figure 5.9 (Contd.) Convergence Pattern of Various Back-propagation Network Architectures

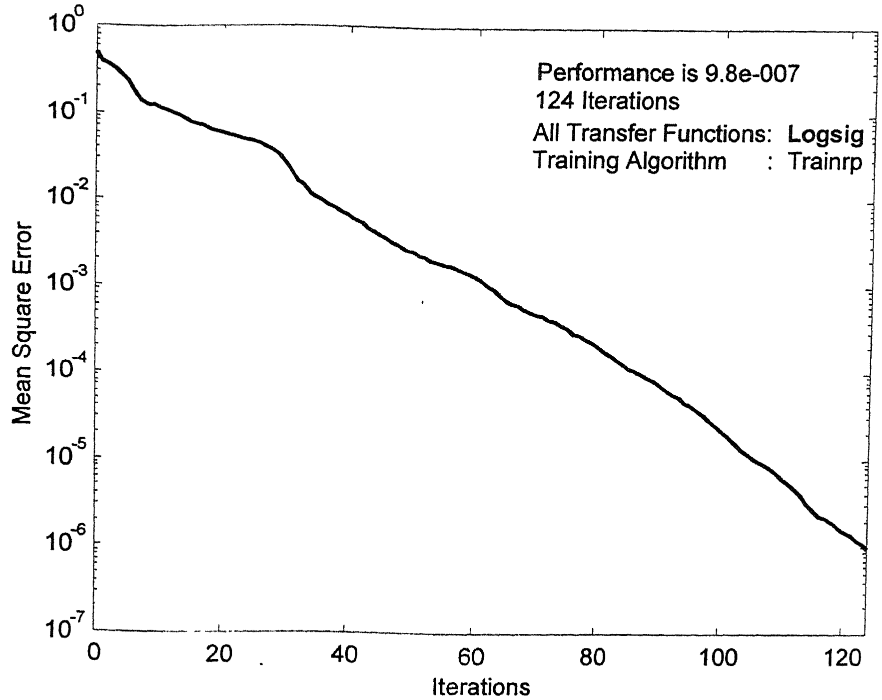


(e) Architecture 20-17-6

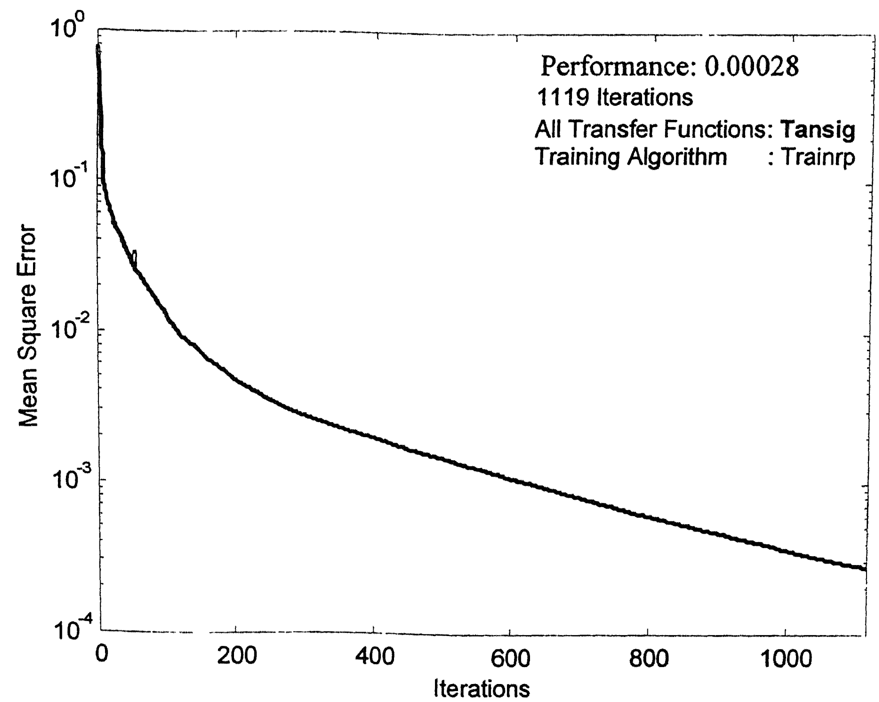


(f) Architecture 20-17-8-6

Figure 5.9 (Contd.) Convergence Pattern of Various Back-propagation Network Architectures



(g) Architecture 20-20-8-6



(h) Architecture 20-20-8-6 with Transfer Functions 'TANSIG'

Figure 5.9 (Contd.) Convergence Pattern of Various Back-propagation Network Architectures

Table 5.5 Results of Backpropagation Architecture, 20-20-8-6, Transfer Function: LOGSIG, Trainig Algorithm: TRAINRP

Fault	Target	Sample Number									
		1	2	3	4	5	6	7	8	9	10
No Fault	1	1.0000	1.0000	1.0000	0.9998	0.9998	0.9999	1.0000	0.9999	1.0000	1.0000
	0	0.0001	0.0002	0.0000	0.0001	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0001	0.0000	0.0000	0.0001	0.0016	0.0001	0.0003	0.0024	0.0001	0.0001
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mass Unbalance	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1	0.9736	0.9961	0.9982	0.9997	0.9988	0.9995	0.9999	0.9999	0.9999	0.9995
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0047	0.0076	0.0151	0.0061	0.0021	0.0035	0.0021	0.0090	0.0006	0.0012
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bearing Inner Race	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0002
	1	0.9998	0.9999	1.0000	0.9409	0.9999	1.0000	1.0000	0.9650	1.0000	1.0000
	0	0.0000	0.0029	0.0000	0.0010	0.0000	0.0000	0.0008	0.0061	0.0000	0.0066
	0	0.0006	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Ball Spin in Bearing	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0000	0.0001	0.0000
	1	0.9996	0.9993	0.9993	0.9993	0.9993	0.9993	0.9998	0.9996	0.9993	0.9999
	0	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0000	0.0001	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Bearing Outer Race	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0002	0.0559	0.0008	0.0014	0.0002	0.0029	0.4839	0.0132	0.0870	0.0061
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1	1.0000	0.9995	0.9999	0.9998	0.9999	0.9998	0.9996	1.0000	0.9999	0.9997
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Cocked Rotor	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.0000	0.0000	0.0000	0.0000
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0001
	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0001	0.0000
	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Table 5.6 Performance

	Architectures						
Faults	**20-8-6**	**20-10-6**	**20-12-6**	**20-17-6**	**20-15-9-6**	**20-17-8-6**	**20-20-8-6**
No Fault	100%	100%	100%	100%	100%	100%	100%
Mass Unbalance	100%	100%	100%	100%	100%	100%	100%
Bearing Inner Race	100%	100%	100%	90%	100%	90%	100%
Ball Spin in Bearing	100%	100%	100%	100%	100%	100%	100%
Bearing Outer Race	70%	100%	70%	100%	100%	100%	100%
Cocked Rotor	100%	100%	100%	100%	100%	100%	100%
Total	95%	100%	95%	98.33%	100%	98.33%	100%

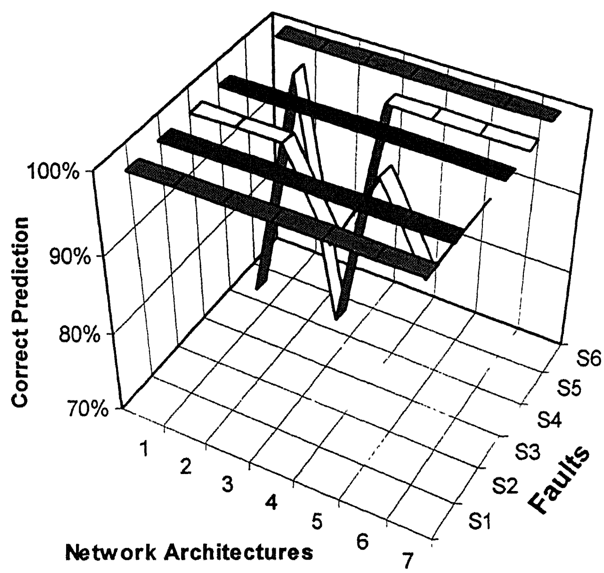


Figure 5.11 Correct Fault Classification (in %) by Various Architectures

LOGSIG (Fig. 3.11(d)) was used as the Transfer Function in the network architectures and TRAINRP was the back-propagation training scheme. Investigations were carried out with other transfer functions (like TANSIG, PURELIN etc.) and training schemes (TRAINGD, TRAINGDA, TRAINGDX, TRAINLM) also. Though all the results are not presented here, a comparison between the convergence pattern for 20-20-8-6 architecture with LOGSIG and TANSIG transfer functions is given in Figs. 5.4(g) and 5.4(h).

5.4 Probabilistic Neural Network Training

A limited study was carried out with Probabilistic Neural Networks also. In a manner similar to that in the case of back-propagation algorithm, six hundred input vectors were used to train the network, while sixty vectors were set aside for testing the network. The Probabilistic Network being a go-nogo kind of gate, gave correct results in all sixty cases. This fact is shown in Fig. 5.12. A major limitation however, of the probabilistic network remains that its dimensions keep increasing with each incoming training vector

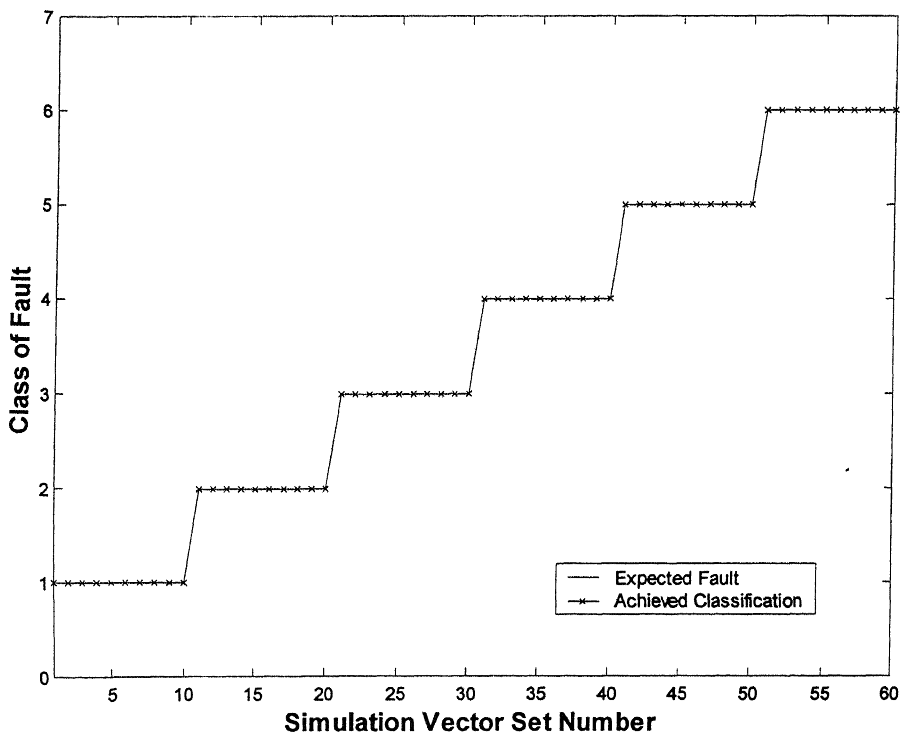


Figure 5.12 Results of Fault Classification by Probabilistic Neural Network

CHAPTER 6

CONCLUSIONS AND SCOPE FOR FUTURE WORK

The objective of this was to carry out multi-point instrumentation on a Fault Simulator Rig and integrate the hardware and software of Virtual Instrumentation with techniques of Artificial Neural Networks for typical fault identification problems in rotordynamics.

It was observed that Virtual Instrumentation technology provides efficient multi-channel display and signal processing capabilities. The hardware was installed in a mutually compatible manner and vibration signals could be acquired from seven different locations on the rotor rig.

Algorithms were developed for on-line Fast Fourier Transformation of the incoming vibration data and automated peak detection from these Fourier plots. These algorithms provided the input data for the Artificial Neural Networks which were based on back-propagation and probabilistic concepts. Both procedures gave excellent results which established the propriety of usage of ANNs for condition monitoring activities in rotordynamics.

However no attempt has been made in this work on quantification of the fault, once it is identified (e.g. estimate of the amount of unbalance, if the fault is identified as unbalance). The focus, presently, was to generate data for healthy and faulty rotor systems and develop a preliminary Neural Network diagnosis frame. It has been found that the testing success in addition to the input and hidden layer architecture.

Further work needs to be carried out to make the ANN shell more realistic by incorporating more number of faults as well as more than one fault in combination with one another. Also investigations have to be conducted to quantify the fault, once the qualitative identification has been made. The present study also does not address the issue, where the faults are in a transitory stage of development.

REFERENCES

1. Childs D., *Turbomachinery Rotordynamics,* John Wiley & Sons, 1993.
2. Bose, N.K., and Liang , P., *Neural Network Foundations, With Graphs, Algorithms, and Applications*, Tata McGraw-Hill, New Delhi,1996.
3. Dimentberg M.F., *Statistical Dynamics of Nonlinear and Time-Varying Systems,* Applied Science Publishers, 1983.
4. Eisenmann R.C. Sr., Eisenmann R.C. Jr., *Machinery Malfunction Diagnosis and Correction,* Prentice-Hall PTR, New Jersey, 1998.
5. Elkodry M.F., Chang K.C., Lee G.C., "Appilcation of neural networks in vibration signature analysis". *Journal of Ennginеering Mechanics*, Vol. 120, No. 2, February 1994.
6. Haykin, S., *Neural Networks. A Comprehensive Foundation*, Mcmillan, New York, 1994.
7. Mayes I.W., "Use of neural network for on-line vibration monitoring". *Proceedings of Institute of Mechanical Engineers,* Vol. 208, 1994.
8. McCormick A.C., Nandi A.K., "Real time classification of rotating shaft loading conditions using artificial neural networks", *IEEE Transactions on Neural Networks,* Vol. 8, No. 3, May 1997.
9. McCulloch, W.S. and Pitts, W.H., "A logical calculus of the ideas imminent in nervous activity", *Bull Math Biophy,* pp. 115-113, 1943.
10. Roa J.S., *Rotor Dynamics*, Third edition, New Age International (P) Limited, New Delhi, 1996.
11. Sohre, J.S., "Turbomachinery Problems and Their Correction", *Standardization and Condition Monitoring Workshop,* Chapter 7, Houston, 1991.
12. Vyas N.S., Jain N., Pandey S., "Fault identification in rotating machinery using neural networks", accepted for publication in the *Journal of Vibration Institute of India*, 2000.
13. Zurada, M.J., "*Introduction to Artificial Neural Systems*", Jaico Publishing House, Delhi, 1999.

A 133713

Date Slip

The book is to be returned on the date last stamped.

....	....
....	....
....	....
....	....
....	....
....	....
....	....
....	....
....	....
....	....
....	....
....	....